یونٹ 1 مسائل کو حل کرنا (Problem Solving)

مختصر تعارف (Short Introduction)

پیچیدہ مسائل کو حل کرنے کا عمل مسئلے کا حل (Problem Solving) کہلاتا ہے۔
یہ یونٹ کسی مسئلے کو سمجھنے اور اسے موثر طریقے سے حل کرنے کے طریقوں کو متعارف کرائے گا۔

حاصلاتِ تعلّم (Students Learning Outcomes)

1۔ مسئلہ حل کرنے کے مراحل۔
- مسئلہ کا تعین
- مسئلہ کی وضاحت
- مسئلہ کو سمجھنا
- مسئلہ حل کرنے کی منصوبہ بندی
- مسئلے کے حل کی واضح وضاحت
- بہترین حل کا انتخاب

2۔ فلو چارٹ (Flowchart)
- فلو چارٹ کی وضاحت
- مسئلہ کو حل کرنے کے لیے ایک فلو چارٹ کی اہمیت
- فلو چارٹ کے لیے لوازم کا تعین
- فلو چارٹ کی علامات کا استعمال
- نمونے کے مسائل کے لیے فلو چارٹ کی ڈرائنگ کا مصرف

3۔ الگورتھم (Algorithm)
- الگورتھم کی وضاحت
- مسئلہ حل کرنے میں الگورتھم کے کردار کی وضاحت
- الگورتھم کی تشکیل
- نمونہ کے مسائل کے لئے الگورتھم تحریر کرنا
- الگورتھم کی کارکردگی کا ادراک
- الگورتھم اور فلو چارٹ کے درمیان تفریق

4۔ ڈیٹا ٹیسٹ کرنا (Test Data)
- ٹیسٹ ڈیٹا کے کا تصور
- ٹیسٹنگ کی اہمیت
- ٹیسٹ کیس (Test Case) کی اقسام

5۔ ویری فکیشن اور ویلیڈیشن (Verification and validation)
- ویری فکیشن (Verification) کے تصور کا مفہوم
- ویلیڈیشن (Validation) کے تصور کی تفہیم

6۔ غلطی کی شناخت اور اس کی اصلاح و درستی
- جانچ کے لیے ٹریس ٹیبل (Trace Table) کا استعمال
- ٹیسٹنگ کے لیے غلط ٹیسٹ کے اعداد و شمار کا استعمال

1.1 مسئلہ حل کرنے کے مراحل (Problem Solving Steps)

کسی مسئلہ کو حل کرنے کے لیے ایک منظم طریقہ کار پر عمل کرنا ضروری ہے۔ مندرجہ ذیل میں ہم مختلف مراحل پر بات کریں گے جن پر عمل کر کے ہم کسی مسئلے کو منظم طریقے سے حل کر سکتے ہیں۔

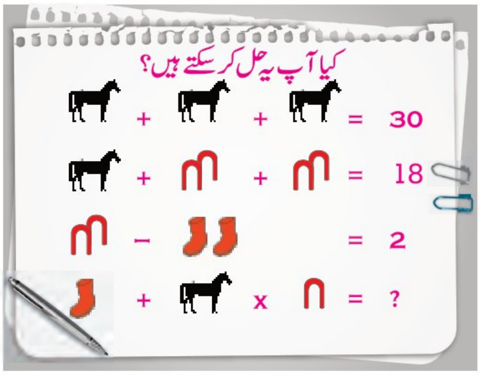
1.1.1 مسئلہ کا تعین (Defining a Problem)

ایک واضح مسئلہ میں کوئی غلط فہمی نہیں ہوتی۔ تمام بنیادی باتیں واضح طور پر متعین کی گئی ہوتی ہیں اور یہ واضح طور پر منزل رکھتا ہے۔ یہ سمجھنے اور حل کرنے میں آسان ہوتا ہے۔ جب ایک مسئلہ بیان کیا جاتا ہے تو سب سے پہلے ہمیں یہ دیکھنے کی ضرورت ہے کہ آیا مسئلے کو واضح طور پر بیان کیا گیا ہے یا نہیں۔ اگر مسئلہ واضح نہ ہو تو ہم ذیل میں دیے گئے طریقوں میں سے کسی ایک طریقہ کو اختیار کر کے مسئلہ کا تعین بآسانی کر سکتے ہیں۔

مسئلے کا پس منظر معلوم کرنا: ہم ان حالات و واقعات کو جاننے کی کوشش کرتے ہیں جن کی وجہ سے مسئلہ پیدا ہو رہا ہوتا ہے۔ اس طریقے سے ہم اس کی شناخت کر سکتے ہیں اس سے یہ بھی جاننے میں مدد ملتی ہے کہ ایک اچھا حل کیسا ہوگا۔ ہم کیونکر حل کو ماپنے (Measure) کے قابل ہوں گے۔

اندازہ لگانا: عدم دستیاب معلومات کا اندازہ لگانے کی کوشش کی جاتی ہے یہ اندازہ ہمارے ماضی کے تجربے کی بنیاد پر ہو سکتا ہے۔

تصویر بنانا: مسئلے کی اچھی طرح وضاحت کرنے کے لیے ایک تصویر بنا سکتے ہیں اور اس سے غیر واضح معلومات کو اخذ کی جا سکتی ہیں۔



شکل 1-1 میں اس کا تصویری تصور پیش کیا گیا ہے

البرٹ آئن سٹائن نے کہا تھا اگر مجھے ایک گھنٹا سیارے کو بچانے کے لیے دیا جائے تو میں 59 منٹ مسئلے کی وضاحت اور ایک منٹ اسے حل کرنے پر خرچ کروں گا۔

تصاویر الفاظ سے کہیں زیادہ بولتی ہیں۔

1.1.2 مسئلے کو سمجھنا (Understanding a Problem)

ضروری ہے کہ مسئلے کو حل کرنے سے پہلے اِسے سمجھا جائے۔ مثال کے طور پر ایک پہیلی کا جواب اسے مکمل طور پر سمجھنے کے بعد ہی دیا جا سکتا ہے ایک مسئلے کو واضح سمجھنے سے اس کو حل کرنے میں آسانی پیدا ہوتی ہے وقت اور وسائل کو بچانے میں مدد ملتی ہے۔ مندرجہ ذیل پانچ سوالات کو سامنے رکھ کر ایک مسئلے کو سمجھا جاتا ہے۔ ایک مسئلے کی تقسیم میں پانچ ڈبلیو (5ws) کی پہچان شامل ہوتی ہے۔

شکل 1-2 مسئلے کا حل

(i) what (کیا)، (ii) who (کون)، (iii) when (کب)،
(iv) where (کہاں)، (v) why (کیوں)۔

مسئلے کا تجزیہ دیئے گئے مسئلے کو سمجھنے میں مدد دیتا ہے ۔ یہ بنیادی عناصر ہیں جو ہمیں دیے گئے مسئلے کے حل کی طرف لے کر جاتے ہیں۔ مثال کے طور پر درج ذیل مسئلہ پر غور کریں:

> فرض کریں کہ آپ کے کلاس ٹیچر آپ کو اپنے سکول میں اُن طلبہ کی فہرست تیار کرنے کا کام دیتا/دیتی ہے جن کے نام کا آغاز حرف A سے شروع ہوتا ہو۔ تمام سکول کے طلبا کی ایک حروف تہجی کے لحاظ سے ڈائریکٹری تیار کرنے کے لیے فہرست کی ضرورت ہے۔ اس ٹاسک کو مکمل کرنے کے لیے صرف ایک ہفتہ مقرر ہے۔ ہم ذیل میں دی گئی مسئلے کے بیان میں پانچ ڈبلیو (5ws) کی شناخت کر کے اس مسئلے کا تجزیہ کر سکتے ہیں۔

کیا **(What)**: ان طلبہ کی فہرست جن کا نام حرف A سے شروع ہوتا ہو۔

کون **(Who)**: طلبہ

کیوں **(Why)**: طالب علموں کی ڈائریکٹری تیار کرنا

کب **(When)**: ایک ہفتے میں

کہاں **(Where)**: سکول میں

شکل **2.1** مسئلے کی علامتی نمائندگی ظاہر کرتی ہے۔ یہاں سرخ بتی ایک مسئلہ کو پیش کرتی ہے پیلی بتی اس کے تجزیے کو پیش کرتی ہے اور سبز بتی اس کے حل کو پیش کرتی ہے۔ یہ ظاہر کرتا ہے کہ مسئلے کا تجزیہ ہمیں اس کے حل کے قریب لے جاتا ہے۔

طلبہ کے دو یا تین گروپ بنا دیے جائیں اور ہر گروپ کو طلبہ کے ناموں کی دو مختلف فہرستیں فراہم کی جائیں ایک فہرست طلبہ کے ریاضی کے مضمون کے نمبروں پر مشتمل ہو جبکہ دوسری فہرست میں فزکس کے مضمون کے نمبر ہوں ہر گروپ کو مندرجہ ذیل فہرستیں تیار کرنی ہیں۔

ریاضی میں سب سے زیادہ نمبر لینے والے 5% طلبہ

فزکس میں سب سے زیادہ نمبر لینے والے 5% طلبہ

طلبہ جو ان دونوں مضامین میں 90% سے زیادہ نمبر رکھتے ہیں اس مسئلہ کے لیے 5ws کی شناخت کریں۔

1.1.3 حل کی منصوبہ بندی (Planning a solution)

کسی مسئلے کا تجزیہ کرنے کے بعد ہم ایک منصوبہ تیار کرتے ہیں یہ ہمیں ایک مسئلہ کے حل کی طرف لے جا سکتا ہے۔ اس مرحلے پر مسئلہ حل کرنے کے لئے درست حکمت عملی کی بھی ضرورت ہے۔ چند حکمت عملیاں پیش ہیں:

شکل 3-1 کامیابی کے لیے منصوبہ بندی

کامیابی کے لیے منصوبہ بندی

تقسیم کریں اور فتح حاصل کریں (Divide and Conquer)

یہ حکمت عملی پیچیدہ مسئلے کو چھوٹے مسئلوں میں تقسیم کرتی ہے۔

اندازہ لگائیں، جانچیں اور بہتر بنائیں (Guess, Check and Improve)

ڈیزائنر مسئلے کے حل کا اندازہ لگاتا ہے اور پھر حل کی درستی کو چیک کرتا ہے۔ اگر حل توقعات کے مطابق نہیں ہے تو وہ حل کو تبدیل کرتا/کرتی ہے۔ حل کو بہتر کرنا ایک تکراری عمل ہے۔

ایکٹ اِٹ آؤٹ (Act it out)

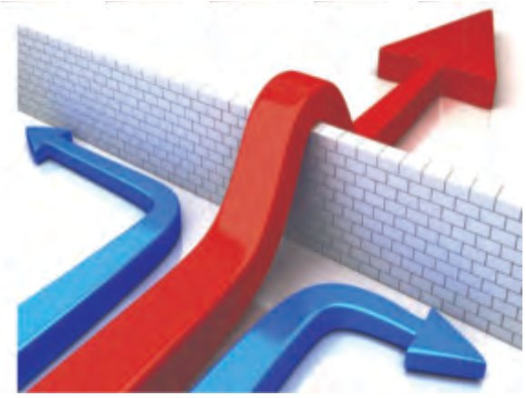
اس حکمت عملی میں ڈیزائنر کاموں کی فہرست تیار کرتا ہے۔ اس کے بعد وہ اس ہر کام کو سرانجام دیتا ہے۔

نمونہ (Prototype)

یہ تکنیک حل کی ایک شاندار نمائندگی کرتی ہے اگرچہ یہ آخری حل نہیں ہوتا تاہم ڈیزائنر کی مدد کر سکتا ہے۔ یہ حل کے اہم اجزاء کو سمجھنے میں مدد دیتی ہے۔

حکمت عملی کے انتخاب کا انحصار مسئلے کی نوعیت پر ہوتا ہے۔ یہ بہت اہم ہے کہ ایک حکمت عملی کسی مسئلے کو حل کرنے میں دوسری حکمت عملی سے بہتر ہو۔ کسی بھی حکمت عملی کا انتخاب مسئلے کی نوعیت پر ہوتا ہے۔

1.1.4 کینڈڈ سلوشن کی وضاحت (Defining Candid Solutions)



شکل 4-1 ایک مسئلے کے ایک سے زیادہ حل

لفظ کینڈڈ سلوشن غیر منصوبہ بندی کا حوالہ دیتا ہے۔ مثال کے طور پر اگر آپ سے پوچھا جائے کہ آپ کے سکول میں ایسے طلبہ کی کتنی تعداد ہے جو کرکٹ کھیل سکتے ہیں؟ آپ اندازہ اس طرح سے کر سکتے ہیں کہ اپنی کلاس میں طلبہ شمار کریں جو کرکٹ کھیل سکتے ہیں۔ پھر اس کو سکول میں موجود تمام کلاسوں سے ضرب دیں آپ کے پاس ان لڑکوں کی تعداد آ جائے گی جو کرکٹ کھیل سکتے ہیں۔ آپ کا جواب اس طریقے سے کینڈڈ سلوشن ہوگا۔ کرکٹ کے کھلاڑیوں کی درست تعداد جاننے کے لیے آپ کچھ اور طریقے بھی اختیار کر سکتے ہیں جیسا کہ ہر جماعت میں جانا یا اساتذہ سے ڈیٹا/مواد حاصل کرنا۔ کوئی کسی کے بارے میں کسی وقت بھی سوچ سکتا ہے۔ ایک کنڈڈ سلوشن وقت بچانے میں مددگار ثابت ہوتا ہے۔ شکل 1.4 میں مختلف طریقے دکھائے گئے ہیں ایک خاص جگہ تک پہنچے کے لیے (جہاں تک پہنچا جا سکتا ہے دیوار کو پار کر کے یا اس کی ایک طرف سے گزر کر) اور جو ایک حل آپ سوچیں وہ کام کرنے کا کنڈڈ سلوشن ہوگا۔ یہ ضروری نہیں ہے کہ مسئلے کا کنڈڈ سلوشن حقیقت میں اس کا حل ہو۔

1.2

آپ کا ہدف اپنے ہم جماعتوں کی اوسط لمبائی معلوم کرنا ہے آپ ایک اندازے سے اس کا جواب دیں اور اس کا درست نتیجہ نکالنے کا طریقہ بھی بتائیں۔ ایک طریقہ یہ بھی ہے کہ انچ ٹیپ کا استعمال کیا جائے۔ ٹیپ سے لمبائی کے مطابق نشان لگائیں اور پھر اس ٹیپ سے پیمائش پڑھیں۔ ہر طالب علم کی لمبائی لکھنے کے بعد آپ حساب لگا سکتے ہیں کہ آپ کے کمرہ جماعت میں تمام طالب علموں کی اوسط اونچائی کتنی ہے۔ یا آپ ایک اندازے سے بھی ایک طالب علموں کی لمبائی جان سکتے ہیں جیسے کسی چیز کے ذریعے بھی لمبائی بتائی جا سکتی ہے، جیسا کہ ایک کتاب۔ فرض کریں کہ آپ کی درسی کتاب کی لمبائی 8 سینٹی میٹر (8cm) ہے۔ آپ دیوار پر کتاب کی اونچائی سے نشان لگا سکتے ہیں۔ آپ اس کتاب کی پیمائش کو متعدد بار استعمال کر کے دیوار پر 8 سینٹی میٹر کے پیمانے سے نشان لگا سکتے ہیں۔ پھر ایک نشان زد دیوار کے سامنے کسی طالب علم کو کھڑا کر کے اس کے قد و قامت کا اندازہ لگایا جا سکتا ہے۔

1.1.5 بہترین حل کا انتخاب (Selecting the Best solution)

شکل 1-5 حل کی سطحیں

کبھی کبھی ہم ایک سے زیادہ حل تلاش کرتے ہیں اور ان میں سے بہتر کا انتخاب کرتے ہیں۔ مثال کے طور پر فرض کریں کہ آپ کے سکول کے تمام طالب علموں کے نام ایک ویب سائٹ پر موجود ہیں اور آپ کو ایک خاص نام تلاش کرنے کا کہا جاتا ہے آپ یہ مسئلہ مندرجہ ذیل طریقوں کو استعمال کرتے ہوئے حل کر سکتے ہیں:

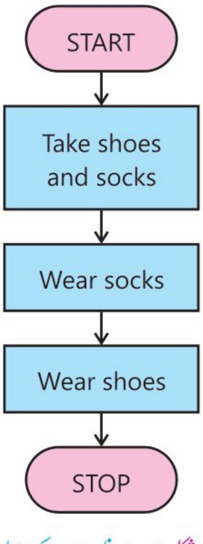
1۔ آپ ویب سائٹ پر موجود سب ناموں کو ایک ایک کر کے دیکھیں یہاں تک کہ نام آپ کو مل جائے یا فہرست ختم ہو جائے۔

2۔ ان ناموں کا پرنٹ حاصل کریں اور مطلوبہ نام تلاش کریں۔

3۔ سارے نام کاپی کریں اور انھیں ایکسل شیٹ میں ڈالیں ان کو حروفِ تہجی کے اعتبار سے ترتیب دیں۔ ایک مرتب شدہ فہرست میں تلاش نسبتاً آسان ہوتی ہے۔

4۔ صرف Ctrl+F کو دبائیں جب فہرست ویب براؤزر پر موجود ہو۔ آپ خودکار طریقے سے تلاش کرنے کے لیے نام لکھ سکتے ہیں۔ اس کے علاوہ کچھ دوسرے حل بھی ہو سکتے ہیں۔ ہم ایک ایسے حل کی شناخت کرتے ہیں جس میں ہمیں کم سے کم اقدامات اٹھانے پڑیں یا یہ دوسرے حل سے وہ زیادہ موثر لگے۔

1.2 فلو چارٹ (Flowchart)



شکل 1-6 فلو چارٹ کی مثال

فلو چارٹ ایک مسئلے کو حل کرنے کے اقدامات میں زیادہ مددگار تصور کیا جاتا ہے۔ ذیل میں ہم فلو چارٹ پر تفصیل سے بات کریں گے۔

1.2.1 تعریف (Defination)

فلو چارٹ کسی مسئلے کے حل کے مراحل کو تصوری شکل میں پیش کرتا ہے۔ ہم ہر قدم پر علامتیں استعمال کر سکتے ہیں اور یہ علامتیں پروسیسنگ کے بہاؤ میں تیروں کے نشانات سے جڑی ہوئی ہیں (شکل 1.6) جرابوں کے ساتھ جوتے پہننے کا ایک سادہ مسئلے کا فلو چارٹ دکھاتی ہے یہ نا صرف اس کے اہم مراحل ظاہر کرتی ہے بلکہ ایک عمل کو مکمل کرنے کا طریقہ بھی بتاتی ہے۔ ایک فلو چارٹ مسئلے کے

حل میں موجود مراحل پر بات چیت کرنے کے لیے بھی استعمال کیا جاتا ہے۔

1.2.2 مسئلے کے حل میں فلو چارٹ کی اہمیت(Importance of Flowcharts in Problem Solving)

مسئلے کو حل کرتے ہوئے فلو چارٹ حل کی منصوبہ بندی کے لیے استعمال کیا جا سکتا ہے۔ اگر فلو چارٹ پہلے ہی موجود ہو تو ہم آسانی سے سمجھ سکتے ہیں کہ مسئلہ کیسے حل کیا گیا ہے۔ متن کے بجائے تصویری طور پر کسی حل کو دیکھنا زیادہ موثر ہے تصویری اظہار اس بات کی تصدیق کو بھی آسان بناتا ہے کہ حل درست ہے یا نہیں۔ اس کے علاوہ دوسرے لوگوں سے ایک مسئلے کے حل کے بارے میں بات کرنے کا بھی یہ ایک اچھا طریقہ ہے۔

1.2.3 فلو چارٹ کے لوازم کا تعین(Determining Requirements for a Flowchart)

ایک فلو چارٹ میں ہم ان پٹ، آؤٹ پٹ، فیصلہ سازی اور پروسیسنگ کا استعمال کرتے ہیں۔ یہ تصورات نیچے بیان کیے گئے ہیں۔ ان پٹ کا مطلب یوزر (صارف) سے ڈیٹا لینا اور اس کو پروسیس کرنا ہے۔ اس کے بعد کچھ آؤٹ پٹ دینا جیسا کہ شکل 1.7 میں دکھایا گیا ہے۔ ایک فلو چارٹ بنانے کے لیے مندرجہ ذیل ضروریات کا جاننا ضروری ہوتا ہے۔

اِن پٹ(Input)

اس کا مطلب یوزر (صارف) سے ڈیٹا لینا ہے۔ یہ جاننا ضروری ہے کہ کتنا اور کس طرح کے ان پٹ کی ضرورت ہے۔

پروسیسنگ(Processing)

ایک فلو چارٹ پروسیسنگ کے مختلف مراحل پر مشتمل ہوتا ہے پروسیسنگ کے مراحل کو حساب کتاب کرنے اور ان کے نتائج کو سٹور کرنے کے لیے استعمال کیا جاتا ہے۔ اس میں کسی مقدار میں کمی بیشی یا دو مقداروں کو جمع یا ضرب یا تقسیم کرنا شامل ہے۔

شکل 7-1 ان پٹ سے آؤٹ پٹ کی طرف بہاؤ

فیصلہ سازی(Decision Making)

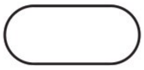
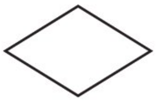
اس بات کا تعین کرنا کہ آیا ایک بیان درست ہے یا غلط ہے، اور اس مطابق مناسب اقدامات کرنا فیصلہ سازی کہلاتا ہے۔

آؤٹ پٹ(Output)

آؤٹ پٹ کا استعمال معلومات کو ظاہر کرنے کے لیے کیا جاتا ہے اور عموماً یہ معلومات عمل در آمد کے نتائج پیش کرتی ہیں۔

1.2.4 فلو چارٹ کی علامات(Flowchart Symbols)

فلو چارٹ علامتوں اور متن کے ذریعے ایک عمل کو واضح طور پر بیان کرتا ہے۔ یہ خاص اشکال استعمال کرتا ہے جو ایک عمل میں موجود مختلف اقدامات ظاہر کرتی ہے۔ لکیریں اور تیر بہاؤ کو ظاہر کرتے ہیں۔ ٹیبل میں 1.1 فلو چارٹ میں استعمال ہونے والی علامات میں سے کچھ کو دکھایا گیا ہے۔

	فلولائن (Flow Line)	یہ کسی فلو چارٹ میں مرحلے کے بہاؤ (Flow) کا تعین کرنے کے لیے استعمال کیا جاتا ہے۔
	ٹرمینل (Terminal)	یہ فلو چارٹ کے آغاز اور اختتام کی طرف اشارہ کرتا ہے۔
	عمل (Processing)	یہ مقدار (Value) کے تبدیل ہونے کے آپریشن کی نمائندگی کرتا ہے۔
	فیصلہ سازی (Decission)	یہ ایک مشروط بیان ظاہر کرتا ہے جو اس بات کا تعین کرتا ہے کہ راستوں میں سے کون سا راستہ اختیار کیا جائے۔ آپریشن عام طور ایک ہاں/نہیں کا سوال یا ایک صحیح/غلط ٹیسٹ ہے۔
	ان پٹ/آؤٹ پٹ (Input/Output)	یہ صارف سے ڈیٹا کے ان پٹ کے طور پر لینے کی نشاندہی کرتا ہے یا صارف کو نتائج دکھاتا ہے۔
	کنیکٹر (Connector)	اگر ایک فلو چارٹ صفحے پر پورا نہیں آتا تب ہم ایک کنیکٹر (Connector) کے ذریعے فلو چارٹ کے حصوں کو ملا دیتے ہیں۔

ٹیبل 1.1 فلو چارٹ کی علامات

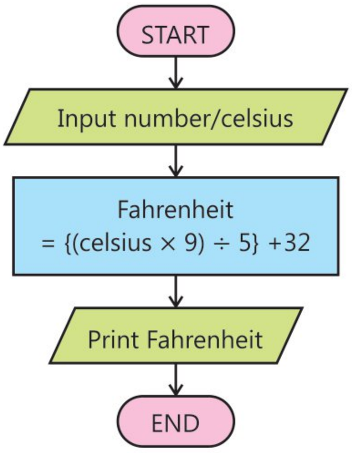
1.2.5 فلو چارٹ کی مثالیں (Examples of Flowchart)

اس حصے میں ہم فلو چارٹ استعمال کرتے ہوئے مسائل کو حل کرنا سیکھیں گے۔ فلو چارٹ میں ان پٹ کا مطلب یہ ہے کہ کسی ان پٹ ڈیوائس (Device) کا استعمال کرتے ہوئے ڈیٹا کو کمپیوٹر میں میموری (Memory) تک پہنچا دیا جائے۔

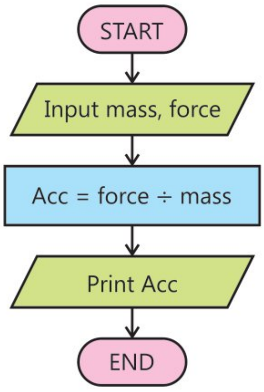
میموری میں موجود ڈیٹا کو مخصوص نام دیا جاتا ہے۔ ڈیٹا کے دیے گئے نام سے اس تک رسائی حاصل کی جاتی ہے تاکہ اسے کسی فارمولا میں استعمال کیا جائے اور کوئی نتیجہ اخذ کیا جائے۔ نتائج کو بھی کمپیوٹر کی میموری میں رکھ دیا جاتا ہے۔ جنھیں بعد میں کسی آؤٹ پٹ ڈیوائس پر دیکھا جاسکتا ہے۔

1- درجہ حرارت کو سینٹی گریڈ سے فارن ہائیٹ کی اکائی میں تبدیل کرنا (To convernt Celsius to Fahrenheit Temperature)

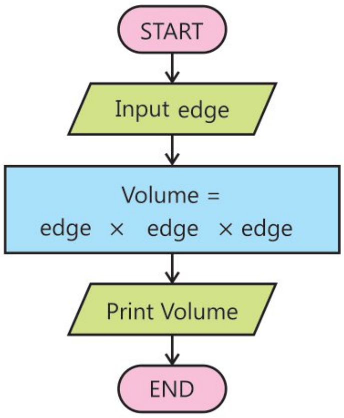
ابتدا میں یہ فلو چارٹ ظاہر کرتا ہے کہ آغاز کے بعد صارف کمپیوٹر کو عددی ڈیٹا دے گا۔ جسے کمپیوٹر اپنی میموری میں سیلسئس (Celcius) کے نام سے محفوظ رکھے گا۔ اگلے مرحلے میں سیلسئس کو فارن ہائیٹ میں تبدیل کرنے والے فارمولا میں اس کو استعمال کیا جائے گا۔ نتیجے کو فارن ہائیٹ میں محفوظ کر لیا جائے گا۔ فارن ہائیٹ کے نام سے محفوظ شدہ قیمت کو کسی آؤٹ پٹ ڈیوائس پر دکھایا جاتا ہے۔ اس مرحلے کے بعد فلو چارٹ ختم ہو جاتا ہے۔



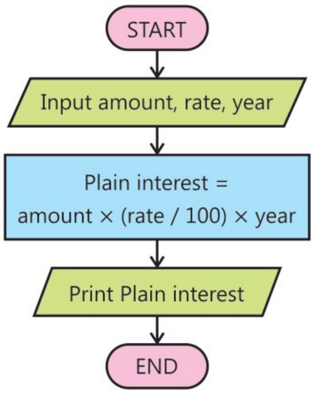
2- کسی حرکت کرتی چیز کا اسراع معلوم کرنا جس کی کمیت اور لگائی گئی قوت پہلے سے دی گئی ہو۔



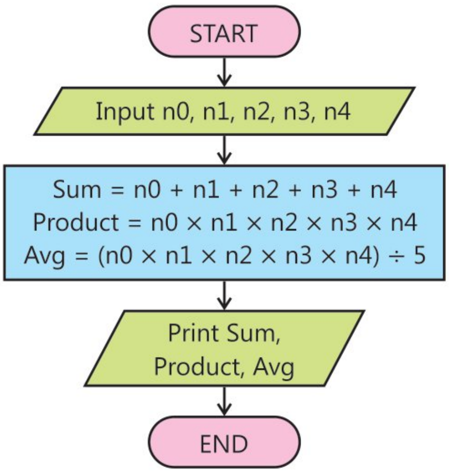
3- کیوب کا حجم تلاش کرنا



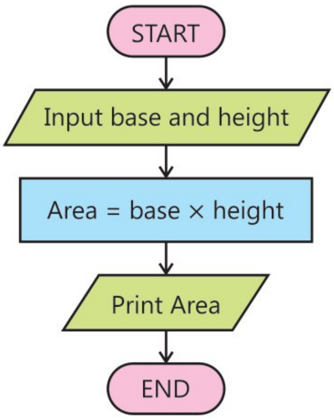
4- ایک رقم پر پلین انٹرسٹ (Plain Interest) معلوم کرنا



5- پانچ اعداد کا مجموعہ، ضرب اور اوسط معلوم کرنا



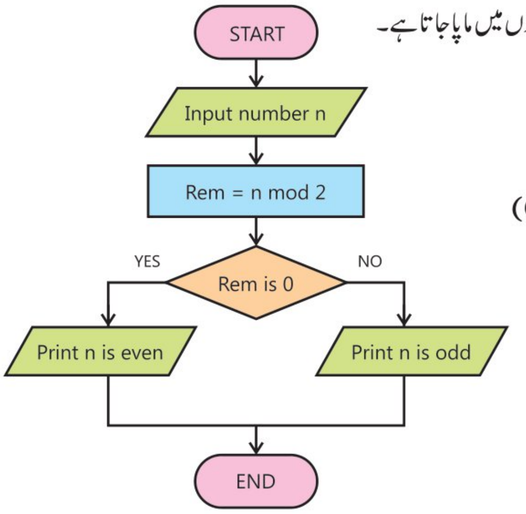
6- متوازی الاضلاع کا رقبہ معلوم کرنا



فلو چارٹ میں مشروط بہاؤ (Conditional Flow in Flowchart)

نیچے دیے گئے فلو چارٹ میں ہم مطالعہ کریں گے کہ مراحل کے درمیان بہاؤ کس طرح سے مشروط ہوتا ہے۔ شرط درست ہونے پر بہاؤ مختلف ہوگا اس بہاؤ سے جس میں شرط کا نتیجہ غلط ہوگا۔ ایک شرط کو ہمیشہ درست یا غلط معنوں میں ماپا جاتا ہے۔

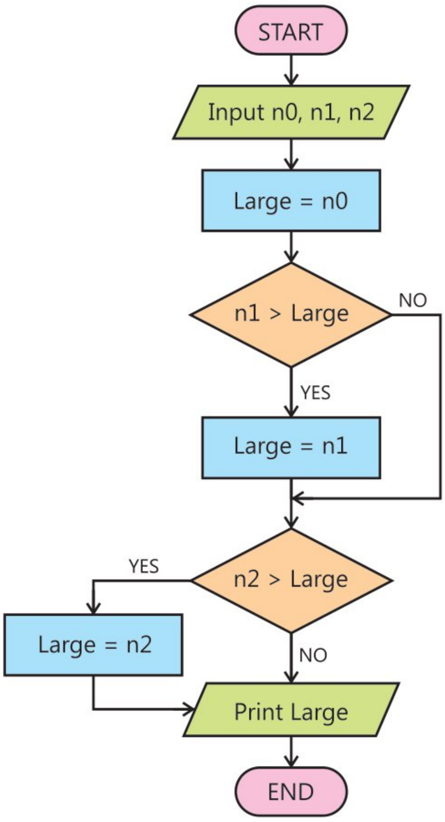
7- دی گئی رقم کے بارے میں جفت (Even) یا طاق (Odd) معلوم کرنا۔



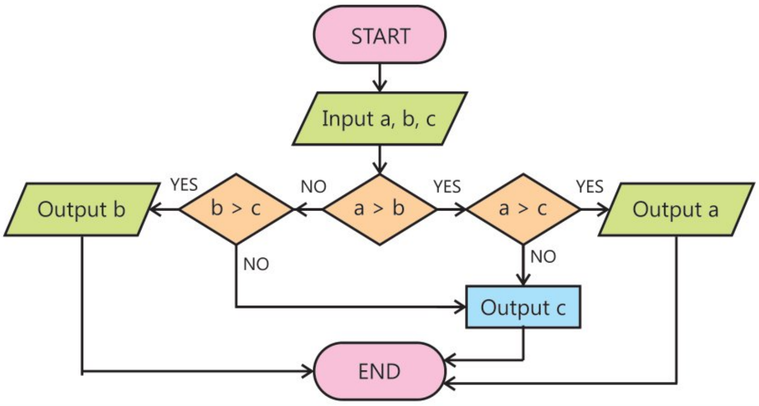
مندرجہ بالا فلو چارٹ میں ابتدائی مرحلہ کے بعد صارف ایک عددی قیمت مہیا کرتا ہے۔ جسے کمپیوٹر میموری میں (n) کے نام سے محفوظ کیا جاتا ہے۔ پھر اس قیمت کو 2 سے تقسیم کیا جاتا ہے اور باقی بچ جانے والی رقم کو (Rem) کے نام سے دوبارہ کمپیوٹر میموری میں محفوظ کر لیا جاتا ہے۔ بقیہ رقم معلوم کرنے کے لیے موڈ (Mod) کا فنکشن استعمال کیا گیا ہے۔ یہ جاننے کے لیے کہ آیا (n) مکمل طور پر (2) سے تقسیم ہوسکتا ہے یا نہیں، ہم مشروط آپریشن سر

انجام دیتے ہیں۔ یہ کام (Rem) کو زیرو سے موازنہ کرنے کے بعد ہوتا ہے اگر معلوم ہو جائے کہ Rem کی قیمت (0) ہے تو مشروط آپریشن اس کی قیمت کو درست بتاتا ہے۔ جس کا مطلب یہ ہوا کہ (n) ایک جفت ہے کیونکہ یہ دو پر مکمل تقسیم ہوسکتا ہے۔ دوسری صورت میں اگر Rem کی قیمت صفر نہیں ہے تو مشروط آپریشن اس کی قیمت کو غلط قرار دیتا ہے۔ جس سے پتہ چلتا ہے کہ (n) ایک طاق عدد ہے۔

8- دیے گئے تین غیر مساوی اعداد میں سب سے بڑا عدد معلوم کرنا



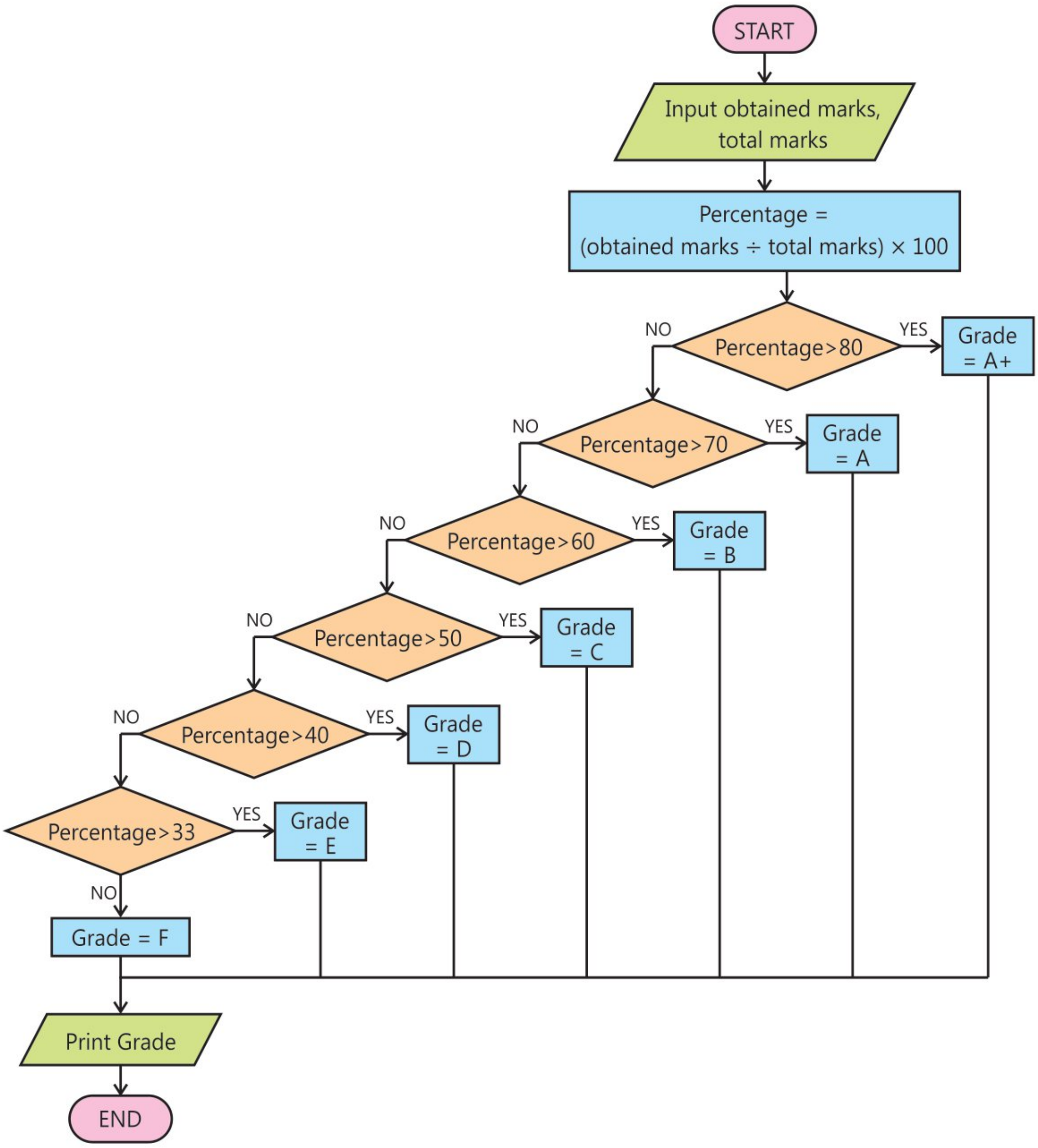
مندرجہ ذیل فلو چارٹ تین اعداد a, b اور c میں سے بڑی قیمت کو معلوم کرنے کا ایک اور طریقہ کار بتاتا ہے۔



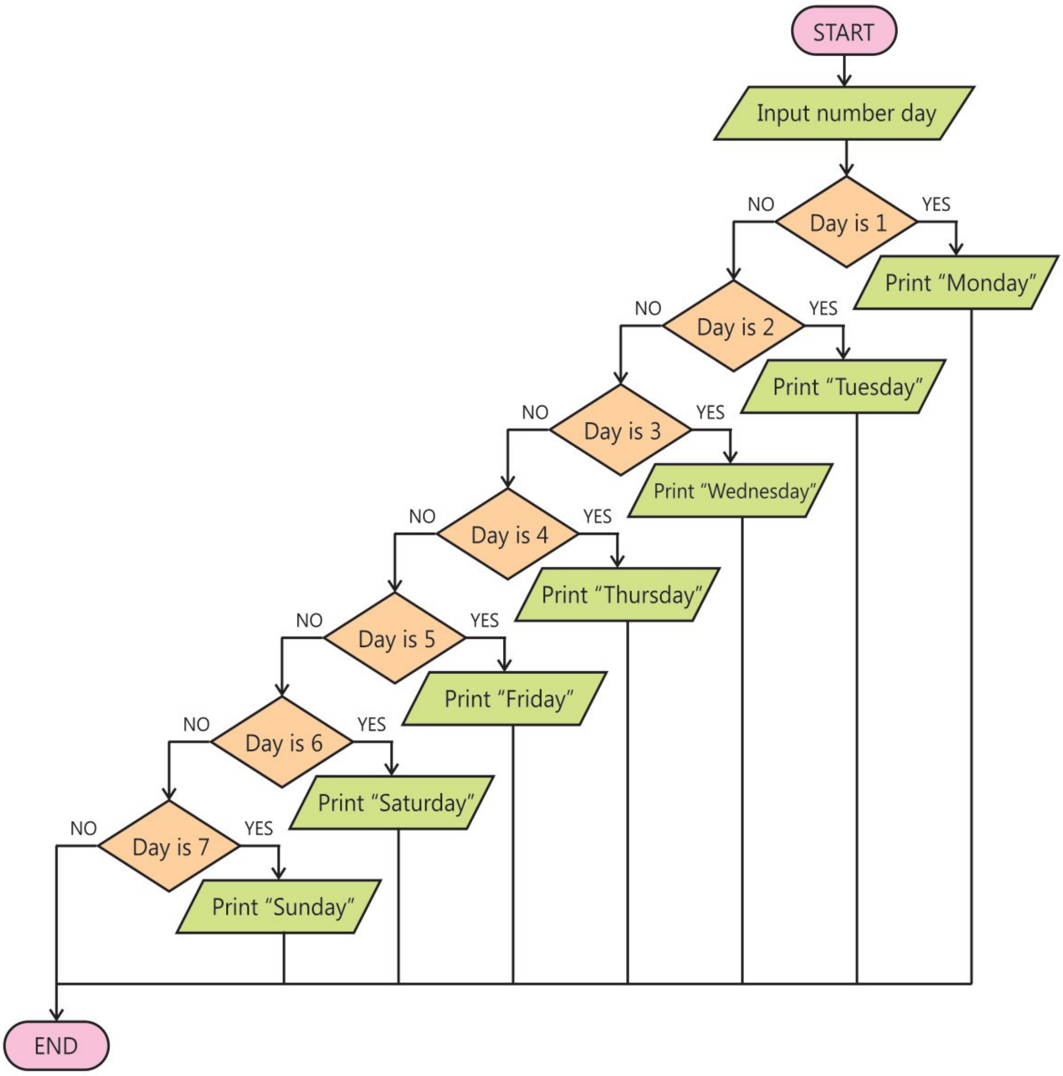
1.3

تین نمبروں میں سے بڑے نمبر کو تلاش کرنے کے لیے دو مختلف فلو چارٹ دیے ہیں۔ آپ ان دونوں کا موازنہ کریں اور اپنے نتائج کو لکھیں اور اپنے کلاس ٹیچر سے گفتگو کریں۔

9-کسی مضمون کا گریڈ معلوم کرنا جب کہ مجموعی نمبر اور حاصل کردہ نمبر دیے گئے ہوں۔



10- دیے گئے نمبر سے ہفتے کے دن کا نمبر معلوم کرنا جبکہ سوموار سے اتوار تک کے دنوں کے لیے 1 سے 7 تک نمبر ہیں۔

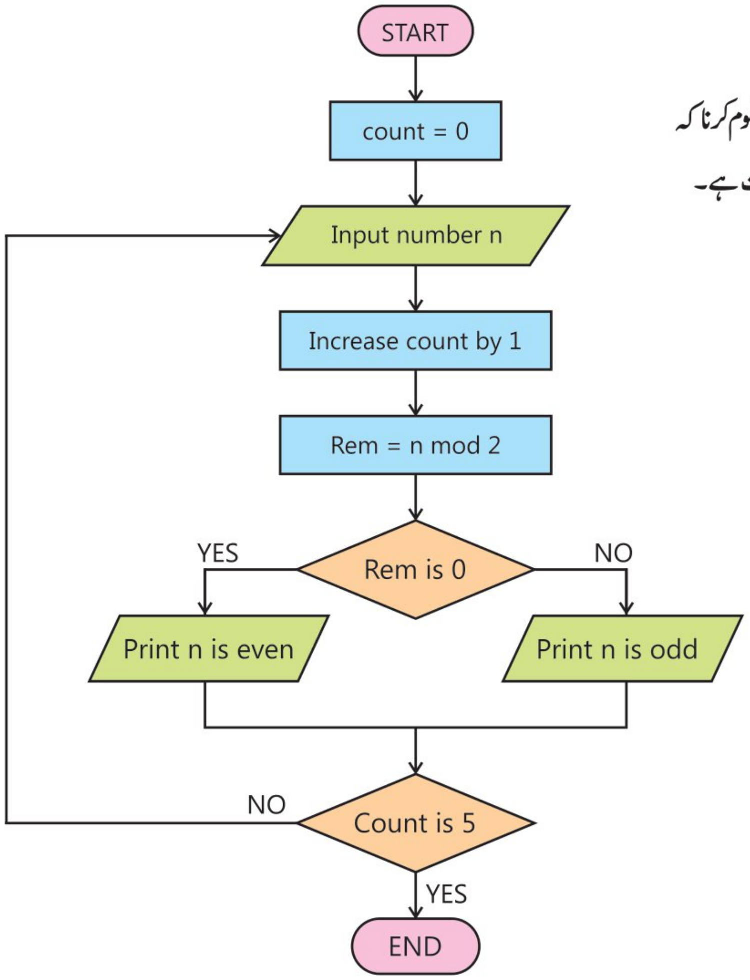


1.4

مندرجہ بالا فلو چارٹ میں صارف نے 1 سے 7 تک کوئی نمبر مہیا کرنا ہوتا ہے۔عین ممکن ہے کہ وہ نمبر 1 سے چھوٹا یا 7 سے بڑا بھی ان پٹ دے۔اس صورت میں ہمیں فلو چارٹ کو اس طرح سے تبدیل کرنا ہے کہ صارف کو اس میں غلط ان پٹ کے ہونے سے متعلق پیغام مل جائے۔

فلو چارٹ میں مراحل کو کئی بار دہرانا (Repeating Steps in Flowcharts)

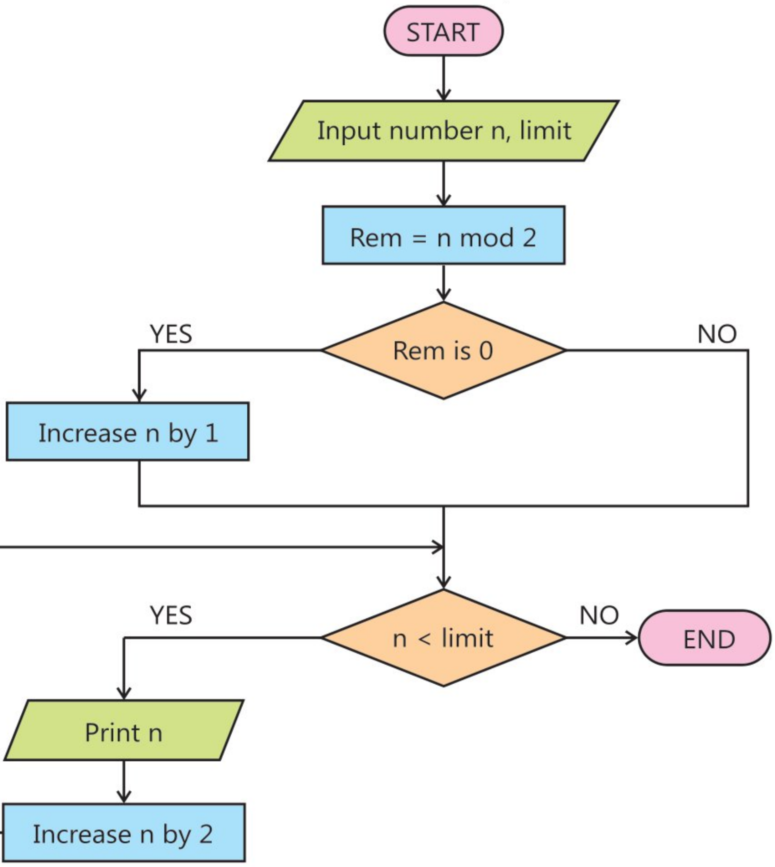
مندرجہ بالا تمام فلو چارٹس میں بہاؤ صرف اوپر سے نیچے کی طرف ہے جبکہ یہ بھی ہوسکتا ہے کہ بہاؤ کسی پچھلے مرحلے کی جانب دوبارہ چلا جائے۔ بالخصوص جب ہمیں مراحل کو دہرانا پڑے۔ مندرجہ ذیل فلو چارٹس میں مراحل کو دہرانے کا عمل دکھایا گیا ہے۔



11- پانچ قیمتوں کو ایک ایک کر کے معلوم کرنا کہ کون سی قیمت طاق ہے اور کون سی جفت ہے۔

مندرجہ بالا فلو چارٹ میں ایک قدر کاؤنٹ (Count) کے نام سے متعارف کرائی گئی ہے۔ جس کا مقصد قیمتوں کو شمار کرنا ہے۔ ابتداً (Count) کی قیمت زیرو (0) ہے جس میں ایک ایک کر کے اضافہ کیا جاتا ہے اور (Count) کی قیمت پانچ ہو جاتی ہے۔ تو اس سے معلوم ہوتا ہے کہ ہم نے پانچ قیمتوں کو جفت یا طاق کی شکل میں معلوم کر لیا ہے۔

12- دو اعداد کے درمیان طاق اعداد معلوم کرنا۔



مندرجہ بالا فلو چارٹ کو جفت نمبروں کے حساب سے تبدیل کریں اس مسئلے کو حل کرنے کے ایک سے زیادہ طریقے بھی ہو سکتے ہیں۔ آپ اپنے فلو چارٹ کا اپنے ہم جماعتوں کے بنائے ہوئے فلو چارٹ سے موازنہ کریں۔

1.3 الگورتھم (Algorithm)

الگورتھم سیکھنا کمپیوٹر سائنس کی تعلیم کا بنیادی جزو ہے۔ اس حصے میں ہم الگورتھم کو تفصیل سے بیان کریں گے۔

1.3.1 تعریف (Defination)

الگورتھم مسئلہ حل کرنے کے مراحل کے مجموعے کا نام ہے اسے فطری زبان میں لکھا جاتا ہے۔ اس لیے یہ قابل فہم ہوتا ہے جیسے: چائے تیار کرنے کے مسئلے کو حل کرنے کے لیے مندرجہ ذیل مراحل طے کرنا ہوں گے:

(a)	Start	۔سٹارٹ	مرحلہ 1
(b)	Take a Kettle	۔کیتلی لیں	مرحلہ 2

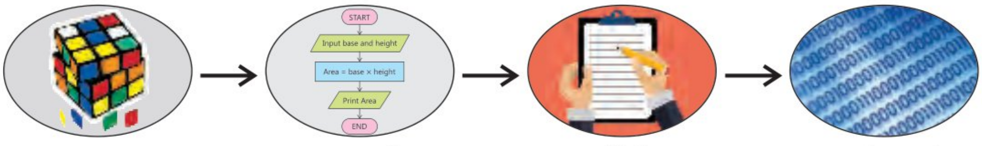
(c)	Pure Water in it	۔اس میں پانی ڈالیں	مرحلہ 3
(d)	Put the Kettle on fire	۔کیتلی کو آگ پر رکھیں	مرحلہ 4
(e)	Add Sugar and milk	۔چینی اور دودھ ڈالیں	مرحلہ 5
(f)	Wait till it boils	۔اس کے ابلنے کا انتظار کریں	مرحلہ 6
(g)	Remove the Kettle from fire	۔کیتلی آگ سے اتار لیں	مرحلہ 7
(h)	End	۔اختتامیہ	مرحلہ 8

اوپر دیے گئے مراحل کا سیٹ چائے بنانے کا الگورتھم پیش کرتا ہے۔کمپیوٹر کی مدد سے ہم کئی مسائل کا حل تلاش کر سکتے ہیں۔اس مقصد کے لیے ہم سب سے پہلے ایک الگورتھم وضع کرتے ہیں جسے بعد میں کمپیوٹر کے لیے ہدایت میں تبدیل کر دیا جاتا ہے۔عام طور پر الگورتھم کو ان پٹ مہیا کی جاتی ہے۔ جسے وہ پرسیس (Process) کرنے کے بعد آؤٹ پٹ فراہم کرتا ہے۔جیسا کہ شکل میں دکھایا گیا ہے۔

لفظ الگورتھم عربی مشہور عربی سائنس دان محمد ابن موسیٰ الخوارزمی کے نام سے لیا گیا ہے۔

1.3.2 الگورتھم مسئلہ حل کرنے میں اہم کردار ادا کرتا ہے

یہ مسئلہ حل کرنے والے کو مرحلہ وار رہنمائی فراہم کرتا ہے۔ یہ حل کو مکمل طور پر بیان کرتا ہے۔عموماً کمپیوٹر پروگرامر سب سے پہلے ایک الگورتھم ہی لکھتا ہے۔ پھر اس کو کمپیوٹر کی زبان میں تبدیل کر دیتا ہے۔بعض اوقات کمپیوٹر پروگرامر سب سے پہلے فلو چارٹ بناتا ہے۔اور پھر اس کو الگورتھم میں تبدیل کرتا ہے۔



شکل 8-1 الگورتھم کا کردار

1.3.3 الگورتھم کی تشکیل

الگورتھم لکھنے کے لیے مختلف علامات استعمال کی جاتی ہیں ہم ٹیبل 1.2 میں دی گئی علامات کو استعمال کرتے ہوئے الگورتھم لکھیں گے۔

سٹارٹ (Start)	یہ کسی الگورتھم کے ابتدائی نقطہ کو ظاہر کرتی ہے۔ ہر الگورتھم کا ایک ابتدائی نقطہ ہوتا ہے۔
ان پُٹ (Input)	یہ علامت کسی یوزر سے ان پُٹ لینے کے لیے استعمال ہوتی ہے جس کو بعد میں کمپیوٹر کی میموری میں محفوظ کیا جاتا ہے۔
سیٹ (Set)	یہ کسی بھی مواد کو نام دینے کے لیے استعمال ہوتی ہے اس کو کسی بھی متغیرہ (Variable) کی قیمت تبدیل کرنے کے لیے بھی استعمال کیا جاتا ہے۔
اِف ایلس (if-else)	اس کا استعمال کسی کنڈیشن کو جانچنے کے لیے کیا جاتا ہے۔ مثال کے طور پر کنڈیشن (a<b) درست یا غلط ہوسکتی ہے اگر یہ درست ہو تو if والا حصہ چلے گا اور اگر کنڈیشن غلط ہوئی تو else والا حصہ چلے گا مثلاً اگر a=5، b=7 تو if(a<5)set c to 10 else set c to 20 else کو لکھنا ضروری نہیں۔
گوٹو (Goto)	اس کا استعمال کنٹرول کو پروگرام کے ایک حصے سے دوسرے حصے میں منتقل کرنے کے لیے کیا جاتا ہے یہ عام طور پر لوپ کی جگہ پر متبادل کے طور پر استعمال ہوتی ہے۔
آؤٹ پُٹ (Output)	یہ علامت اقدار دیکھانے کے لیے استعمال ہوتی ہے۔
سٹاپ (Stop)	یہ ایک الگورتھم کے اختتامی نقطہ کو ظاہر کرتی ہے۔

ٹیبل 1-2 الگورتھم میں استعمال ہونے والی علامات

1.3.4 الگورتھم کی مثال

1۔ پانچ نمبروں کو جمع کریں ضرب دیں اور اوسط معلوم کریں۔

Step 1. Start — مرحلہ نمبر 1۔ آغاز

Step 2. Input numbers, $n0, n1, n2, n3, n4$ — مرحلہ نمبر 2۔ پانچ نمبرز $n0, n1, n2, n3, n4$ ان پٹ کے طور پر لیں۔

Step 3. Set *sum* to $n0 + n1 + n2 + n3 + n4$. — مرحلہ نمبر 3۔ Sum نام کے ویری ایبل میں تمام نمبرز کا مجموعہ محفوظ کریں۔

مرحلہ نمبر 4۔ Product نام کے ویری ایبل میں تمام نمبرز کا حاصلِ ضرب محفوظ کریں۔

Step 4. Set *product* to $n0 \times n1 \times n2 \times n3 \times n4$

Step 5. Set *average* to $\frac{n0+n1+n2+n3+n4}{5}$ — مرحلہ نمبر 5۔ Average نام کے ویری ایبل میں تمام نمبرز کی اوسط محفوظ کریں۔

Step 6. Output *sum, product, average* — مرحلہ نمبر 6۔ Sum، Product اور Average کی قیمتیں سکرین پر دکھائیں۔

Step 7. End — مرحلہ نمبر 7۔ اختتامیہ

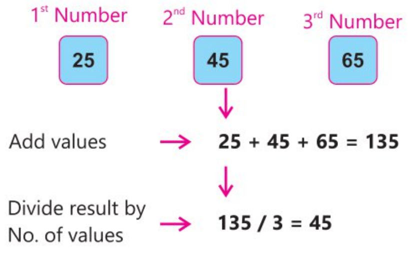
اس الگورتھم میں مرحلہ نمبر 1 الگورتھم کا آغاز دکھاتا ہے۔ مرحلہ نمبر 2 سے معلوم ہوتا ہے کہ صارف 5 عددی قیمتیں فراہم کرتا ہے اور وہ اسے کمپیوٹر میموری میں n0, n1, n2, n3 اور n4 کے ناموں سے محفوظ کر لیتا ہے۔ مرحلہ نمبر 3 تمام ان پٹ قیمتوں کا خلاصہ ظاہر کرتا ہے اور جمع (Sum) کے نام سے کمپیوٹر میموری میں نتائج محفوظ کرتا ہے۔ مرحلہ نمبر 4 تمام نمبروں کے ضرب کے حساب کے طور پر استعمال کیا جاتا ہے۔ اور اس کا نتیجہ ضرب (Product) کے نام سے محفوظ کیا جاتا ہے۔

مرحلہ نمبر 5 میں اوسط پانچ نمبروں کا حساب کرنے کے لیے فارمولہ لگایا جاتا ہے اور نتیجہ کو اوسط (Average) کے نام سے محفوظ کرلیا جاتا ہے۔ مرحلہ نمبر 6 کے بالترتیب مرحلہ نمبر 3،4 اور 5 کے نتائج دکھاتا ہے۔ مرحلہ نمبر 7 الگورتھم کے اختتامیہ کو ظاہر کرتا ہے۔

1.6

شکل تین نمبروں 25، 45 اور 65 کی اوسط معلوم کرنے کی ایک سادہ مثال ہے۔ ان مقرر کردہ قیمتوں کو شمار کرنے کے ایک الگورتھم لکھیں اور ان میں اوسط قیمتیں دکھائیں۔

(نوٹ کریں کہ اس صورت میں صارف سے کوئی ان پٹ لینے کی ضرورت نہیں۔)



Example of calculating Average

2۔ کسی حرکت کرتے ہوئے جسم کا ایکسلریشن معلوم کرنا جب اس کی کمیت (Mass) اور استعمال ہونے والی قوت (Force) دی گئی ہو۔

Step 1. Start
Step 2. Input numbers, *mass*, *force*
Step 3. Set *acceleration* to $\frac{force}{mass}$
Step 4. Output *acceleration*
Step 5. End

مرحلہ نمبر 1۔ آغاز
مرحلہ نمبر 2۔ کمیت (Mass) اور قوت (Force) صارف سے ان پٹ لیں۔
مرحلہ نمبر 3۔ Acceleration کو Force/Mass کے برابر رکھیں۔
مرحلہ نمبر 4۔ Acceleration کو سکرین پر دکھائیں۔
مرحلہ نمبر 5۔ اختتام

شکل 1-9 ایک ہی کمیت (mass) والی چیز پر زیادہ قوت (Force) لگائی جائے تو زیادہ اسراع (Acceleration) پیدا ہوگا۔

3۔ مکعب (Cube) کا حجم (Volume) معلوم کرنا

Step 1. Start
Step 2. Input number, *side*
Step 3. Set *volume* to $side \times side \times side$.
Step 4. Output *volume*
Step 5. End

مرحلہ نمبر 1۔ آغاز
مرحلہ نمبر 2۔ Side کو ان پٹ کے طور پر لیں۔
مرحلہ نمبر 3۔ Volume کو SidexSidexSide کے برابر شمار کریں۔
مرحلہ نمبر 4۔ Volume کو سکرین پر ظاہر کریں۔
مرحلہ نمبر 5۔ اختتامیہ

1.7

اوپر دیے گئے الگورتھم کو تبدیل کرتے ہوئے سلنڈر اور دائرہ کا حجم معلوم کریں۔ دائرہ کا حجم معلوم کرنے کے لئے فارمولہ $\frac{4}{3} \times \pi r^3$ ہے۔ جہاں r سے مراد ریڈیس (radius) ہے۔ سلنڈر کا حجم معلوم کرنے کے لیے فارمولہ $\pi r^2 h$ ہے۔ جہاں (r) ریڈیس اور (h) اس کی اونچائی ہے۔

4- متوازی الاضلاح کا رقبہ تلاش کریں

Step 1. Start
Step 2. Input numbers, *base, height*
Step 3. Set *area* to $base \times height$
Step 4. Output *area*
Step 5. End

مرحلہ نمبر 1۔ آغاز
مرحلہ نمبر 2۔ Base, Height کو ان پٹ کے طور پر لیں۔
مرحلہ نمبر 3۔ Area کو Height x Base کے برابر شمار کریں۔
مرحلہ نمبر 4۔ Area کو سکرین پر ظاہر کریں۔
مرحلہ نمبر 5۔ اختتامیہ

1.8

اُوپر دیے گئے الگورتھم کو مثلث، معین (Rhombus) اور مستطیل (trapezium) کا رقبہ معلوم کرنے والے الگورتھم میں تبدیل کریں۔

5۔ دیے گئے تین نمبروں میں سے بڑا نمبر معلوم کرنا۔

Step 1. Start
Step 2. Input numbers, $n0, n1, n2$
Step 3. Set *large* to $n0$
Step 4. if $n1 > large$ Set *large* to $n1$
Step 5. if $n2 > large$ Set *large* to $n2$
Step 6. Output *large*
Step 7. End

مرحلہ نمبر 1۔ آغاز
مرحلہ نمبر 2۔ نمبرز n0، n1، n2 صارف سے اِن پٹ لیں۔
مرحلہ نمبر 3۔ ویری ایبل Large میں نمبر $n0$ محفوظ کریں۔
مرحلہ نمبر 4۔ اگر n1 ویری ایبل Large سے بڑا ہے تو Large میں n1 محفوظ کریں۔
مرحلہ نمبر 5۔ اگر n2 ویری ایبل Large سے بڑا ہے تو Large میں n2 محفوظ کریں۔
مرحلہ نمبر 6۔ Large کو سکرین پر دیکھائیں۔
مرحلہ نمبر 7۔ اختتامیہ

اس الگورتھم میں مرحلہ نمبر 1 الگورتھم کے آغاز کی نشاندہی کرنا ہے مرحلہ نمبر 2 یوزر سے مطالبہ کرتا ہے کہ وہ تین (3) نمبرز ان پٹ دے جو کہ کمپیوٹر کی میموری میں بالترتیب n0، n1 اور n2 کے ناموں سے محفوظ ہوجاتے ہیں۔ یہ الگورتھم شروع میں یہ فرض کرتا ہے کہ سب سے بڑی قدر n0 ہے اور اس

کا دوسری اقدار سے موازنہ کرتا ہے مرحلہ نمبر (3) یہ ظاہر کرتا ہے کہ $n0$ میں محفوظ شدہ قدر کو large نام کے ویری ایبل میں محفوظ کیا جاتا ہے۔ مرحلہ نمبر (4) میں Large کا موازنہ n1 سے کیا جاتا ہے۔ اگر n1 کی قیمت large کی قیمت سے بڑی ہے تو large میں n1 کی قیمت رکھ لی جاتی ہے۔ اگر n1 کی قیمت large سے کم ہو تو مرحلہ نمبر 4 large کی قیمت پر اثر انداز نہیں ہوتا۔ اسی طرح مرحلہ نمبر 5 large کی قیمت تبدیل کر سکتا ہے اگر n2 کی قیمت large کی قیمت سے بڑی ہو تو مرحلہ نمبر 6 میں large کو آؤٹ پُٹ میں دکھایا گیا ہے۔

6۔ کسی مضمون میں حاصل کردہ نمبروں کی بنیاد پر اس مضمون کو گریڈ دیں۔

مرحلہ نمبر 1۔ آغاز

Step 1. Start

مرحلہ نمبر 2۔ صارف سے حاصل کردہ نمبر اور ٹوٹل نمبر اِن پٹ کے طور پر لیں۔

Step 2. Input numbers, *obtained_marks, total_marks*

مرحلہ نمبر 3۔ نمبروں کا فیصد ($\frac{\text{حاصل کردہ نمبر}}{\text{ٹوٹل نمبر}}$ x100) فارمولا استعمال کرتے ہوئے شمار کریں۔

Step 3. Set *percentage to* $\frac{obtained_marks}{total_marks} \times 100$

مرحلہ نمبر 4۔ اگر اوسط 80 سے زیادہ ہے تو گریڈ میں +A محفوظ کریں۔

Step 4. If $percentage > 80$ Set *grade* to *A+*

نہیں تو

else

اگر اوسط 70 سے زیادہ ہے تو گریڈ میں A محفوظ کریں۔

if $percentage > 70$ Set *grade* to *A*

نہیں تو

else

اگر اوسط 60 سے زیادہ ہے تو گریڈ میں B محفوظ کریں۔

if $percentage > 60$ Set *grade* to *B*

نہیں تو

else

اگر اوسط 50 سے زیادہ ہے تو گریڈ میں C محفوظ کریں۔

if $percentage > 50$ Set *grade* to *C*

نہیں تو

else

اگر اوسط 40 سے زیادہ ہے تو گریڈ میں D محفوظ کریں۔

if $percentage > 40$ Set *grade* to *D*

نہیں تو

else

اگر اوسط 33 سے زیادہ ہے تو گریڈ میں E محفوظ کریں۔

if $percentage > 33$ Set *grade* to *E*

نہیں تو

else

گریڈ میں F محفوظ کریں۔

Set *grade* to *F*.

مرحلہ نمبر 5۔ گریڈ کو سکرین پر دکھائیں۔

Step 5. Output *grade*

مرحلہ نمبر 6۔ اختتامیہ

Step 6. End

7۔ کسی رقم پر انٹرسٹ معلوم کرنا

مرحلہ نمبر 1۔ آغاز

Step 1. Start

مرحلہ نمبر 2۔ رقم، ریٹ، سال کو صارف سے اِن پٹ کے طور پر لیں۔

Step 2. Input numbers, *amount, rate, years*

مرحلہ نمبر 3۔ انٹرسٹ کو (سال × ($\frac{\text{ریٹ}}{100}$ × رقم)) فارمولا استعمال کرتے ہوئے معلوم کریں۔

Step 3. Set *plain_interest* to $\left(amount \times \frac{rate}{100}\right) \times years$

Step 4. Output *plain_interest* — مرحلہ نمبر 4۔ انٹرسٹ کوسکرین پر دکھائیں۔

Step 5. End — مرحلہ نمبر 5۔ اختتامیہ

8۔ درجہ حرارت کو سیلسیئس سے فارن ہائیٹ سکیل میں اور فارن ہائیٹ کو سیلسیئس میں تبدیل کریں۔

Step 1. Start — مرحلہ نمبر 1۔ آغاز

Step 2. Input number, *celsius* — مرحلہ نمبر 2۔ صارف سے سیلسیئس سکیل میں درجہ حرارت اِن پٹ کے طور پر لیں۔

مرحلہ نمبر 3۔ فارن ہائیٹ کو 32+($\frac{9\times \text{سیلسیئس}}{5}$) فارمولا استعمال کرتے ہوئے شمار کریں۔

Step 3. Set *fahrenheit* to $\frac{celsius \times 9}{5} + 32$

Step 4. Output *fahrenheit* — مرحلہ نمبر 4۔ فارن ہائیٹ کو سکرین پر دکھائیں۔

Step 5. Input number, *fahrenheit* — مرحلہ نمبر 5۔ صارف سے درجہ حرارت فارن ہائیٹ سکیل میں ان پُٹ کے طور پر لیں۔

مرحلہ نمبر 6۔ سیلسئس کو 5/9x(32-فارن ہائیٹ) فارمولا استعمال کرتے ہوئے شمار کریں۔

Step 6. Set *celsius* to $(fahrenheit - 32) \times \frac{5}{9}$

Step 7. Output *celsius* — مرحلہ نمبر 7۔ سیلسیئس کو سکرین پر دکھائیں۔

Step 8. End — مرحلہ نمبر 8۔ اختتامیہ

9۔ نمبر n1 سے لے کر نمبر n2 کے درمیان طاق اعداد معلوم کریں (جبکہ n2 نمبر n1 نمبر سے بڑا ہے۔)

Step 1. Start — مرحلہ نمبر 1۔ آغاز

Step 2. Input numbers, $n1, n2$ — مرحلہ نمبر 2۔ نمبرز n1, n2 صارف سے ان پٹ کے طور پر لیں۔

Step 3. if $(n1 \leq n2)${ — مرحلہ نمبر 3۔ اگر n1 نمبر n2 سے چھوٹا یا اِس کے برابر ہے تو

Step 4. if ($n1$ mod 2 equal 0) Output n1 — مرحلہ نمبر 4۔ {اگر n1 کا 2 سے حاصلِ باقی 0 ہو تو n1 کو سکرین پر دکھائیں۔

Step 5. Set $n1$ to $n1 + 1$ — مرحلہ نمبر 5۔ n1 نمبر میں 1 کا اضافہ کردیں۔

Step 6. go to Step 3 — مرحلہ نمبر 6۔ مرحلہ نمبر 3 پر واپس جائیں۔

} — {

Step 7. End — مرحلہ نمبر 7۔ اختتامیہ

1.3.5 الگورتھم کی کارگزاری

ایک مسئلے کو حل کرنے کے لیے ایک سے زیادہ الگورتھم ہو سکتے ہیں اس میں کونسا بہتر ہے اس کا انحصار اس الگورتھم کی کارگزاری پر ہوتا ہے۔ کسی بھی الگورتھم کی کارگزاری دو بنیادوں پر جانچی جاسکتی ہے۔

1۔ مراحل کی تعداد

اگر ایک الگورتھم کم مراحل میں درست نتیجہ دیکھا دیتا ہے تو اس کو زیادہ بہتر اور مؤثر سمجھا جاتا ہے۔

2۔ کمپیوٹر کی میموری کا استعمال

الگورتھم استعمال ہونے والے مواد کو کمپیوٹر کی میموری میں محفوظ کرتے ہیں الگورتھم جو کم جگہ یا میموری استعمال کرے اچھا الگورتھم سمجھا جاتا ہے بنسبت اس الگورتھم کے جو زیادہ میموری استعمال کرے۔ یہ بھی ممکن ہے کہ ایک الگورتھم کم میموری استعمال کرنے اور زیادہ مراحل میں ایک مسئلہ حل کرے جبکہ دوسرا

الگورتھم اسی مسئلے کو کم مراحل میں مگر زیادہ میموری کو استعمال کرتے ہوئے حل کرے تو اس مرحلے پر ہمیں میموری یا مراحل میں سے کسی ایک کا انتخاب کرنا پڑتا ہے جو کہ حالات کی مناسبت پر انحصار کرتا ہے۔

مثال:

ہم دو(2) الگورتھم لیتے ہیں جو ایک ہی مسئلے کو حل کرتے ہیں ایک الگورتھم N مراحل میں مسئلے کا حل دیتا ہے جبکہ دوسرا N^2 مراحل میں اسی مسئلے کو حل کرتا ہے ان دونوں میں سے پہلے والے الگورتھم کو بہتر تصور کیا جاتا ہے۔

مثال:

1 سے 99 تک اعداد کو جمع کریں۔

اس مسئلے کو کیسے حل کیا جائے؟

مختلف ذہنوں میں اس مسئلے کے مختلف حل آسکتے ہیں۔ ایک حل یہ بھی ہوسکتا ہے کہ شروع سے آخر تک نمبروں کو جمع کرتے جائیں! دوسرا حل یہ بھی ہوسکتا ہے کہ آپ جوڑے بنائیں جیسا کہ:

(1+99),(2+98),(3+97),(4+96),.............(49+51) جبکہ ہر ایک جوڑا جمع ہو کر 100 بنتا ہے ہم جوڑوں کی تعداد گن لیتے ہیں اور اس کو 100 سے ضرب دیتے ہیں اور حاصل جواب میں 50 جمع کر کے جواب معلوم کیا جاسکتا ہے۔

تیسرا حل یہ ہے کہ فارمولا $\frac{n(n+1)}{2}$ کو استعمال کریں جہاں پر n آخری نمبر ہے۔ مثلاً اس مسئلے کا حل $\frac{99(99+1)}{2}$ ہوگا۔

اُوپر کی مثال سے ہمیں ایک مسئلے کو حل کرنے کے لیے ایک سے زیادہ طریقے ملتے ہیں۔ اگر انہی الگورتھمز کو ہم کمپیوٹر میں چلاتے ہیں تو یہ مختلف مراحل اور مختلف تعداد میں میموری استعمال کریں گے۔

شطرنج میں 64 خانے ہوتے ہیں اگر ہم ہر خانے پر گندم کے دانے اس طرح رکھیں کہ پہلے خانے پر ایک دانہ اور اس سے اگلے خانے پر اس سے دوگنا جیسا کہ اگلے خانے پر دو دانے اس سے اگلے خانے پر اس خانے سے دوگنا جیسا کہ چار دانے یعنی ہر خانے پر پیچھے والے سے دوگنا دانے رکھتے جائیں تو پورے شطرنج پر 18,446,744,073,709,551,615 دانے آئیں گے۔

سیکشن نمبر 1.3.4 میں دیئے گئے الگورتھم نمبر 9 کا مندرجہ ذیل الگورتھم سے موازنہ کر کے بتائیں کہ کون سا الگورتھم بہتر ہے۔ یہ بات قابل غور ہے کہ دونوں ایک ہی مسئلے کو حل کر رہے ہیں۔ مندرجہ بالا سوال کو حل کرنے کے لیے n1 اور n2 کی قیمت فرض کر لیں اور دونوں الگورتھمز میں مراحل کی تعداد گن لیں۔

مرحلہ نمبر 1۔ سٹارٹ/آغاز	Step 1. Start
مرحلہ نمبر 2۔ نمبرز n1، n2 صارف سے ان پٹ کے طور پر لیں۔	Step 2. Input numbers, $n1, n2$
مرحلہ نمبر 3۔ اگر n1 طاق ہے تو n میں ایک کا اضافہ کر دیں۔	Step 3. if $n1$ is odd, set $n1$ to $n1 + 1$
مرحلہ نمبر 4۔ n1 کو سکرین پر دکھائیں۔	Step 4. Output $n1$
مرحلہ نمبر 5۔ سیٹ n1 میں 2 کا اضافہ کریں۔	Step 5. Set $n1$ to $n1 + 2$
مرحلہ نمبر 6۔ اگر n1 نمبر n2 سے چھوٹا ہے تو مرحلہ نمبر 4 پر چلے جائیں۔	Step 6. if $n1 < n2$ go to Step 4
مرحلہ نمبر 7۔ اختتامیہ	Step 7. End

1.3.6 الگورتھم اور فلو چارٹ میں فرق

الگورتھم اور فلو چارٹ کے درمیان فرق صرف ایک کہانی اور ایک فلم کے درمیان فرق کی طرح ہے۔ جیسا کہ ہم پڑھ چکے ہیں کہ فلو چارٹ ایک مسئلے کو حل کرنے کے عمل کی تصویری نمائندگی کرتا ہے لیکن ایک الگورتھم انھیں مراحل کو فطری زبان میں تحریر کرتا ہے۔

فلو چارٹ کے فوائد

- یہ بنانا آسان ہے۔
- مسئلہ کو حل کرنے میں آسانی پیدا کرتا ہے۔
- غلطیوں کی شناخت کرنے میں آسانی پیدا کرتا ہے (اگر کوئی ہے)
- ایک مرحلے سے دوسرے تک کے فلو یا بہاؤ مشاہدہ کرنا آسان ہے۔

فلو چارٹ کے نقصانات

- فلو چارٹ بنانے کے لئے زیادہ وقت درکار ہوتا ہے۔
- ہر مرتبہ فلو چارٹ میں ترمیم آسان نہیں ہوتی۔

- یہ بہت بڑے مسئلے کے لئے مناسب نہیں ہے۔

الگورتھم کے فوائد

- یہ آسانی سے لکھا جا سکتا ہے
- الگورتھم لکھنے کی تکنیک سمجھنا آسان ہے۔
- بڑے مسئلے کو حل کرنے کے لئے الگورتھم مددگار رہوتا ہے۔

الگورتھم کے نقصانات

- موجودہ الگورتھم میں ہر بار ترمیم آسان نہیں ہوتی ہے۔
- ایک مرحلے سے دوسرے مرحلے تک کا فلو/ بہاؤ دیکھنا آسان نہیں ہے
- اگر goto سٹیٹمنٹ کا استعمال کیا گیا ہوتو اغلاط تلاش کرنا مشکل ہو جاتا ہے۔

1.4 ٹیسٹ ڈیٹا

ایک مسئلے کو حل کرنے کے بعد اس کو ٹیسٹ کیا جاتا ہے کہ حل درست ہے یا نہیں اور اس ٹیسٹ کے لیے ہمیں ٹیسٹ ڈیٹا کی ضرورت ہوتی ہے۔ مثال کے طور پر اگر آپ تین نمبروں میں سے بڑا نمبر معلوم کرنے کے لیے الگورتھم کو ٹیسٹ کرنا چاہتے ہیں تو ہم کو تین اقدار کی ضرورت ہوگی۔ یہ اقدار مثبت، منفی یا صفر بھی ہوسکتی ہیں۔ مثلاً

(n0=5, n1=15, n2=3),(n0=27, n1=-6, n2=35), (n0=24, n1=0, n2=11) اس لیے ٹیسٹنگ کا سوچنے کے لیے ہمیں ٹیسٹ ڈیٹا کا سوچنا پڑتا ہے۔

فرض کریں آپ کو ایک خودکار حاضری کا سسٹم ٹیسٹ کرنے کے لیے دیا گیا۔ اس سسٹم میں ایک کیمرہ ہر ایک طالب علم کے آنے جانے کی معلومات ریکارڈ کرتا ہے۔ کیمرہ ایک کمپیوٹر کے ساتھ منسلک ہے جس کی ڈیٹا بیس میں تمام طلبہ کی تصاویر محفوظ ہیں۔ جو ہر بچے کی تصویر کا کمپیوٹر میں محفوظ شدہ تصویر سے موازنہ کرتا ہے اور اگر یہ تصویر کمپیوٹر میں محفوظ شدہ تصویر سے مشابہت رکھے تو اس بچے کی حاضری لگ جاتی ہے۔

آپ اس سسٹم کے لیے ٹیسٹ ڈیٹا فراہم کریں۔ اپنے پوائنٹس اس طرح دیں جو اس مسئلے کے حل کو مزید بہتر بنا سکے۔ آپ مختلف موسموں میں مختلف کپڑے پہننے کے عمل کو بھی اس میں سکتے ہیں دو طلبہ جڑواں بھی ہو سکتے ہیں بالوں کی مختلف کٹ بھی کیمرے کے سامنے مختلف تصویر پیش کر سکتی ہے۔

1.4.1 ٹیسٹنگ کی اہمیت

کسی بھی مسئلے کے حل کے دوران ہونے والی غلطیاں معلوم کرنا بہت ضروری عمل ہے۔ اس سے حل مزید بہتر بنتا ہے۔ اگر ایک شخص کسی مسئلے کا حل بناتا ہے اور دوسرا شخص اس حل کو کاروباری مقاصد کے لیے استعمال کرتا ہے تو اس کا انحصار اس حل کی درستی پر ہوتا ہے۔ مثال کے طور پر اگر ہم کسی کے مالی انتظامات کے لیے کوئی حل لکھتے ہیں اور بنک اس کو استعمال کرنا شروع کر دیتا ہے۔ اب اگر اس حل میں کوئی غلطی نکل آئے تو یقیناً مالی نقصان ہوگا۔ اسی لیے ٹیسٹنگ ایک ضروری مرحلہ ہے۔

> سپیس شٹل ''چیلنجر ایس۔ٹی۔ایس L51'' نے اپنی پرواز 28 جنوری 1986 میں حادثاتی طور پر اڑان بھرنے کے صرف 73 سیکنڈز کے بعد ہی ختم کر دی تھی۔

کسٹمر کو کار دینے سے پہلے اس کو اچھی طرح سے ٹیسٹ کیا جاتا ہے۔ مارکیٹ میں نئی کار لانچ کرنے سے پہلے اس کو روبوٹ ڈرائیور کی مدد سے ٹیسٹ کیا جاتا ہے جو اس کو دیوار سے ٹکراتا ہے تاکہ یہ پتہ کیا جا سکے کہ اس کے ایئر بیگ صحیح طریقے سے کام کر رہے ہیں یا نہیں۔ اس سے ڈیزائنرز کو ایسے زیادہ محفوظ بنانے اور حادثہ کے نتیجے میں ہونے والے نقصان کو کم سے کم کرنے کے لیے نئی تراکیب بھی ملتی ہیں۔ اس طرح ٹیسٹنگ سے کوالٹی کو بہتر کیا جا سکتا ہے۔

1.4.2 ٹیسٹ ڈیٹا کی اقسام

کسی بھی حل کی کوالٹی کو بہتر کرنے کے لیے مکمل اور متوازی ٹیسٹ ڈیٹا بہت اہمیت کا حامل ہوتا ہے۔ ہر حل کو مختلف اقسام کے ٹیسٹ ڈیٹا کی ضرورت ہوتی ہے۔ مندرجہ ذیل میں ٹیسٹ ڈیٹا کی کچھ اقسام بیان کی گئی ہیں۔

درست ٹیسٹ ڈیٹا:

یہ ٹیسٹ ڈیٹا اس طرح کی ان پُٹ پر مشتمل ہوتا ہے جو ایک الگورتھم کے تقاضوں کے عین مطابق ہو۔ اگر ایک الگورتھم 1 سے لے کر 100 تک اقدار لیتا ہے تو 1 سے 100 تک اقدار ہی درست ٹیسٹ ڈیٹا میں آئیں گی۔

نادرست ٹیسٹ ڈیٹا:

یہ ٹیسٹ ڈیٹا الگورتھم کے تقاضوں سے ہم آہنگی نہیں رکھتا۔ یہ اس لیے بھی ضروری ہے کہ دیکھا جائے کہ الگورتھم غلط ان پٹ کے لیے کیا رویہ اپناتا ہے اور صارف کو درست ان پٹ دینے کے لیے کیا پیغام دیتا ہے۔

باؤنڈری ٹیسٹ ڈیٹا:

اس ٹیسٹ ڈیٹا میں ایک حل کو انتہائی اقدار کے لیے ٹیسٹ کیا جاتا ہے۔ مثال کے طور پر انٹرسٹ جاننے کے لیے ہم 0 قدر ان پُٹ دے سکتے ہیں یا بہت بڑی قدر دے سکتے ہیں۔

ڈیٹا کا غلط نمونہ (Wrong Data Format)

یہ بہت دانشمندانہ عمل ہے کہ ایک الگورتھم کو غلط نمونہ کے لیے ٹیسٹ کیا جائے مثال کے طور پر جہاں نمبر ان پٹ کی ضرورت ہو وہاں انگریزی حروف تہجی دے دیئے جائیں۔

عدم دستیاب ڈیٹا

یہ بھی ایک ضروری عمل ہے کہ الگورتھم کو اس کی ضرورت سے کم ان پُٹ دے کر ٹیسٹ کیا جائے مثال کے طور پر اگر ایک الگورتھم صارف سے اس کا ڈرائیونگ لائسنس نمبر مانگتا ہے اور صارف یہ ڈیٹا ان پٹ کرنے کے قابل نہیں ہوتا۔ یہ جاننا بھی ضروری ہے تاکہ دیکھا جائے کہ الگورتھم ایسی صورت حال میں کیسا رویہ اختیار کرتا ہے۔

1.5 ویری فیکیشن اور ویلیڈیشن (Verification and Validation)

1.5.1 ویری فیکیشن

ویری فیکیشن سے مراد اس بات کی تصدیق کرنا ہے کہ حل اسی مسئلے کے لیے ہے جس کو کہ حل کی ضرورت تھی۔ مثال کے طور پر اگر آپ ایک رقم پر کمپاؤنڈ انٹرسٹ جاننا چاہتے ہیں تو یہ سادہ انٹرسٹ نہ ہو بلکہ کمپاؤنڈ انٹرسٹ ہی ہو۔

1.5.2 ویلیڈیشن

ویلیڈیشن سے مراد اس بات کی تصدیق کرنا ہوتا ہے کہ آیا حل درست بھی ہے کہ نہیں۔ مثال کے طور پر اگر آپ کو کسی رقم پر کمپاؤنڈ انٹرسٹ جاننے کے لیے کہا گیا ہے تو جو رقم آپ کے حل نے بتائی ہے آیا وہی درست رقم ہے۔ ان کی تصدیق ویلیڈیشن سے کی جاتی ہے۔

مثال:

فرض کریں آپ کو کہا گیا ہے کہ ایک ایسا الگورتھم لکھیں جو نمبرز کی لسٹ ان پُٹ کے طور پر لے۔ اور اس لسٹ کو ترتیب صعودی میں دیکھائے۔ آپ اپنا الگورتھم لکھ کر اپنے استاد کو جمع کرواتے ہیں آپ کا استاد الگورتھم کو نمبرز کی ایک لسٹ دیتا ہے۔ اگر آپ کا الگورتھم نمبرز کی لسٹ دیکھا دیتا ہے تو یہ ویری فائیڈ الگورتھم کہلاتا ہے اور ایک بار جب الگورتھم کی تصدیق/ ویری فیکیشن ہو جائے تو آپ کا استاد دوسرے سٹیپ/ قدم کی طرف جائے گا۔ اور جو لسٹ آپ

کے الگورتھم نے دیکھائی ہے آیا وہ صعودی ترتیب میں بھی ہے یا نہیں۔ اگر یہ لسٹ صعودی ترتیب میں ہی ہے اور کوئی نمبر بھی غائب نہیں ہے تو یہ الگورتھم ویلیڈیٹ بھی ہو جاتا ہے۔

مثال:

فرض کریں آپ پیزا کی دوکان پر جاتے ہیں اور چکن پیزا کے لیے آرڈر دیتے ہیں۔ آپ اُسے بتاتے ہیں کہ پیزا زیادہ مصالحے دار نہ ہو۔ آپ یہ بھی توقع کرتے ہیں کہ اس کا ذائقہ اچھا ہو۔ جب پیزا آپ کو مل جائے تو آپ مشاہدہ کر سکتے ہیں کہ پیزا چکن پیزا ہی ہے۔ یہ ویری فیکشن کہلاتی ہے۔ جب آپ پیزا کھاتے ہیں تو آپ یہ چیک کرتے ہیں کہ یہ کم مصالحے دار ہے یا نہیں۔ اس کا ذائقہ اچھا ہے یا نہیں یہ ویلیڈیشن کہلاتی ہے۔

1.6 اغلاط کی نشاندہی اور درستی

اگر ایک الگورتھم ویری فیکیشن کے دوران فیل ہو جائے تو اس کے فیل ہونے کے پیچھے عناصر کا پتہ لگانا بہت ضروری ہے۔ تا کہ انہیں درست کیا جا سکے۔ بعض اوقات غلطی لاجیکل ہوتی ہے جس کا مطلب یہ ہوا کہ آپ کا الگورتھم کام کر رہا ہے مگر مطلوبہ نتائج نہیں دے رہا۔

مثال کے طور پر ہمیں سکول کی والی بال ٹیم کے لیے کھلاڑیوں کا انتخاب کرنے کے لیے ایسے طلبہ کی ضرورت ہے جن کا قد 144 سینٹی میٹر سے 164 سینٹی میٹر کے درمیان ہو۔ اس سلسلے کے لئے مندرجہ ذیل الگورتھم ملاحظہ ہوں۔

مرحلہ نمبر 1۔ آغاز

Step 1. Start

مرحلہ نمبر 2۔ Count کو زیرو کے برابر کریں۔

Step 2. Set count to 0

مرحلہ نمبر 3۔ All height کو [154,140,155,164,144,166,160,143] برابر کریں۔

Step 3. Set all_heights to [154, 140, 155, 164, 144, 166, 160, 143]

مرحلہ نمبر 4۔ All height جب موجود ہر height کے لیے

Step 4. For each height in the list all_heights

مرحلہ نمبر 5۔ اگر height بڑی ہو 144 سے اور چھوٹی یا برابر ہو 164 کے تو Count میں ایک جمع کر دیں۔

Step 5. If $height > 144$ and $height \leq 164$ then Set count to count + 1

مرحلہ نمبر 6۔ Count کو سکرین پر ظاہر کریں۔

Step 6. Output count

مرحلہ نمبر 7۔ اختتام

Step 7. Stop

اوپر دیا گیا الگورتھم کام کرتا ہے مگر سارے طلبہ کا شمار نہیں کرتا۔ مرحلہ نمبر 5 پر جان بوجھ کر ایک غلطی رکھی گئی ہے یہاں > علامت کا استعمال کیا گیا ہے جبکہ یہاں پر ≥ علامت ہونی چاہیے تھی۔ اس لیے ایسا طالب علم جس کی قامت 144 سینٹی میٹر ہو گی نہیں گنا جائے گا جو کہ درست نہیں ہے۔ یہ ایک لاجیکل غلطی ہے ہم اس قسم کی غلطی صرف ٹریس ٹیبل (Trace Table) کے استعمال سے ہی پکڑ سکتے ہیں۔

1.6.1 ٹریس ٹیبل(Trace Table)

یہ تکنیک الگورتھمز کو ٹیسٹ کرنے کے لیے استعمال ہوتی ہے تا کہ اس امر کی یقین دہانی کی جاسکے کہ الگورتھم میں کوئی بھی لاجیکل غلطی نہیں ہے عام طور پر اس ٹیبل میں ایک سے زیادہ قطاریں اور ایک سے زیادہ کالم ہوتے ہیں۔ جہاں پر ہر کالم ڈیٹا کا نام ظاہر کرتا ہے۔اور ہر قطار ڈیٹا کی قیمت ظاہر کرتی ہے۔ٹیبل 1.3 سیکشن 1.6 میں دیئے گئے الگورتھم کے لئے بنایا گیا ہے۔خالی خانے سے مراد ہے کہ کوئی تبدیلی نہیں ہے اور۔۔ے مراد ہے کہ ویلیو کا کوئی تعلق نہیں ہے۔مندرجہ ذیل ٹیبل میں پہلے مرحلے میں ڈیٹا ہر کوئی اثر نہیں ہوتا۔مرحلہ نمبر 2 میں Count کو" 0" ویلیو دے دی گئی ہے۔اور مرحلہ نمبر 3 میں All height کا ویری ایبل متعارف کروایا گیا ہے۔مرحلہ نمبر 4 میں Count اور All height ویری ایبلز اثر انداز نہیں ہو رہا مگر height کے ویری ایبل میں 154 محفوظ کر دیا جاتا ہے۔اسی طرح مرحلہ نمبر 5 میں اس کا موازنہ کیا جاتا ہے اور Count کی قیمت میں ایک کا اضافہ کر دیا جاتا ہے۔اگر تو ویلیو مقررہ حد میں ہے تو مرحلہ نمبر 4 اور 5 کو بار بار دہرایا جاتا ہے۔جیسا کہ ٹیبل 1.3 میں دیکھایا گیا ہے۔

	count	all_heights	height
Step 1	--	--	--
Step 2	0	--	--
Step 3		[154, 140, 155, 164, 144, 166, 160, 143]	
Step 4			154
Step 5	1		
Step 4			140
Step 5	1		
Step 4			155
Step 5	2		
Step 4			164
Step 5	3		
Step 4			144
Step 5	3		
Step 4			166
Step 5	4		
Step 4			160
Step 5	5		
Step 4			143
Step 5	5		
Step 6			
Step 7			

ٹیبل 1.3

1.6.2 نادرست ڈیٹا استعمال کرتے ہوئے ٹیسٹنگ۔

اِس قسم کی ٹیسٹنگ کا مقصد یہ ہوتا ہے کہ اِس بات کو یقینی بنایا جا سکے کہ آپ کا الگورتھم غلط ان پٹ کو بھی مثبت انداز میں ہینڈل کرتے ہوئے صارف کو پیغام دیتا ہے کہ اِن پٹ درست نہیں۔ مثال کے طور پر اگر آپ کے الگورتھم کو آپ کی عمر دنوں میں مطلوب ہے اور آپ اس کو اپنی تاریخِ پیدائش اِن پٹ کے طور پر دے دیتے ہیں تو الگورتھم کو اصولی طور پر صحیح نہیں چلنا چاہیے۔ اِس قسم کی ٹیسٹنگ کا مقصد اِس طرح کے معاملات کا سراغ لگانا ہوتا ہے۔ مزید یہ کہ یہ ٹیسٹنگ آپ کے الگورتھم کی کوالٹی بڑھانے میں بھی مددگار ثابت ہوتی ہے۔

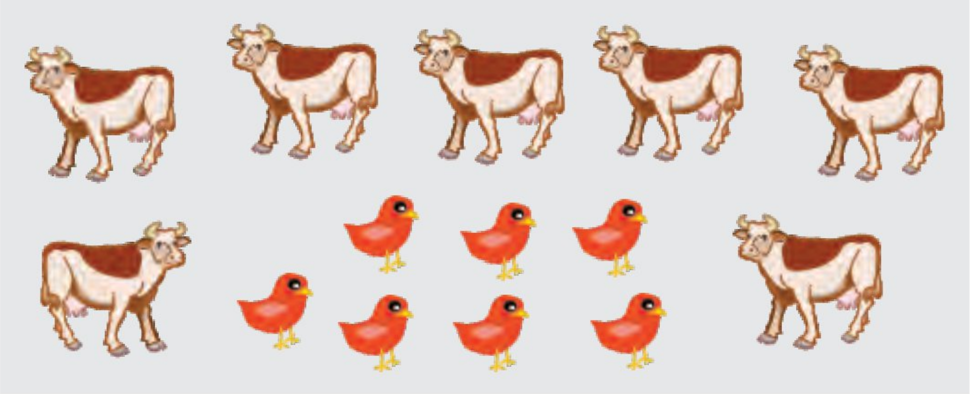
اُوپر دی گئی مثال کے لیے نادرست ڈیٹا ذہین نشین رکھتے ہوئے الگورتھم لکھیں۔ استاد، کلاس کو گروپوں میں تقسیم کرے اور ہر گروپ کو ایک یا ایک سے زیادہ الگورتھم دے۔ طلبہ نادرست اِن پٹ کے لیے صارف کو مناسب پیغام دیں۔

- ایک مسئلے کا تجزیہ اِس کو تیزی سے حل کرنے میں مدد کرتا ہے۔
- اگر چہ کسی مسئلے کے ایک سے زیادہ حل ہو سکتے ہیں مگر بہترین حل وہ ہوگا جس میں کم سے مراحل ہوں اور اس کے لیے کم وقت درکار ہو۔
- الگورتھم ایسے قوانین کا مجموعہ ہوتا ہے جو کسی مسئلے کو حل کرنے کے لیے استعمال میں لائے جاتے ہیں۔
- الگورتھم اِن پٹ لیتا، پروسیس کرتا اور رزلٹ نمایاں کرتا ہے۔
- الگورتھم فیصلہ سازی میں معاون ہے۔
- فلو چارٹ در حقیقت، علامات پر مشتمل ہوتے ہیں جس سے الگورتھم کو تصویری شکل میں پیش کیا جاتا ہے۔
- فلو چارٹس کی علامات یہ ہیں: ان پٹ، آؤٹ پٹ، فیصلہ سازی، تیر، آغاز اور اختتام ہیں۔
- ویلیڈیشن (Validation) سے مراد ہے کہ آپ خاص مسئلے کی نوعیت کے عین مطابق ہیں۔
- ویری فیکیشن (Verification) سے مراد ہے کہ آیا آپ کا حل درست رزلٹ دے رہا ہے یا نہیں۔
- الگورتھم کے جائزے کو ٹیسٹ کرنے کے لیے ٹریس ٹیبل (Trace Table) کی تکنیک استعمال کی جاتی ہے۔

EXERCISE :مشق

1.1 درجہ ذیل سوالات کے جوابات لکھیں۔

1- ایک کھیت میں کچھ گائے اور پرندے موجود ہیں۔ اگر ان کے کل سر 35 اور کل ٹانگیں 110 ہوں تو کہ ان میں گائے اور پرندوں کی تعداد کیا ہوگی؟



2- 'مسئلے کے تجزیے' سے کیا مراد ہے؟ اپنا جواب مثال سے واضح کریں۔

3- الگورتھم کی تعریف کریں اور ایک مسئلے کو حل کرنے میں اِس کے کردار کی وضاحت کریں۔

4- اگر ایک مسئلے کے ایک سے زیادہ کئی حل ہوں تو آپ اِن میں سے بہترین حل کا انتخاب کیسے کریں گے؟ مثالوں کے ساتھ استدلال کریں۔

5- فلو چارٹ کی ضروریات جانچنے کا طریقہ لکھیں۔

6- ٹیسٹ ڈیٹا کی اقسام بیان کریں۔

7- ٹریس ٹیبل سے کیا مراد ہے؟

1.2 درست جواب کا انتخاب کریں۔

1- کس حل کو مناسب الگورتھم پلاننگ (Planning) سے نہیں لکھا جاتا:

(i) تیار شدہ حل (ii) کینڈ حل

(iii) حکمت عملی پر مبنی حل (iv) بہترین حل

2- الگورتھم کا ایک تصویری اظہار ہے:

(i) قالب　(ii) گراف　(iii) فلو چارٹ　(iv) حل

3- فلو چارٹ میں کون سی علامت آغاز اور اختتام کے لیے استعمال کی جاتی ہے۔

(i) ٹرمینل　(ii) کنکٹر　(iii) پروسیس　(iv) ڈائمنڈ

4- ۔۔۔۔۔ کا مطلب ہے کہ آیا مطلوبہ حل موجود ہے یا نہیں!

(i) ویری فکیشن　(ii) الگورتھم　(iii) ویلیڈیشن　(iv) فلو چارٹ

5- ۔۔۔۔۔ قسم کی غلطی کی وجہ سے الگورتھم چل رہا ہوتا ہے مگر درست جواب نہیں دے رہا ہوتا۔

(i) اینڈم ایرر　(ii) لاجیکل ایرر　(iii) سنٹکس ایرر　(iv) رن ٹائم ایرر

1.3 خالی جگہ مکمل کریں۔

1- کسی مسئلے کو حل کرنے سے پہلے اِس کا ۔۔۔۔۔۔ کرنا چاہیے۔

2- الگورتھم ہمیں ۔۔۔۔۔۔ کا ایک مجموعہ فراہم کرتا ہے۔

3- فلو چارٹ مراحل کی ترتیب جانچنے کے مختلف ۔۔۔۔۔ اور ۔۔۔۔۔۔ استعمال کرتا ہے۔

4- فلو چارٹ میں ◇ کی علامت ۔۔۔۔۔۔ کو ظاہر کرتی ہے۔

5- ۔۔۔۔۔۔۔ حل کو ٹیسٹ کرنے کے لیے استعمال کی جاتی ہے۔

1.4 درج ذیل مسائل کے لیے 'فلو چارٹ' بنائیں۔

1- صارف سے دو نمبر N1 اور N2 لیں۔ اور یہ بتائیں کہ N1 نمبر N2 کو مکمل تقسیم کرتا ہے یا نہیں؟

2- صارف سے اِن پٹ کے طور پر کوئی سال لیں۔ بتائیں کہ یہ لیپ (Leap) کا سال ہے یا نہیں؟

3- ایک نمبر صارف سے اِن پٹ لیں اور اِس کا فیکٹوریل (Factorial) شمار کریں۔

4- دو نمبروں کا ایل۔ سی۔ ایم (LCM) معلوم کریں۔

5- اِن پٹ کے طور پر صارف سے ایک نمبر لیں اور اِس کے اجزاء (Factors) معلوم کریں۔

ایک فلو چارٹ بنائیں جو کہ پاکستان موٹروے پر جرمانے کی رقم بتائے۔ جرمانہ مندرجہ ذیل قوائد کے تحت کیا جاتا ہے۔ صارف صرف کوڈ اِن پٹ کے طور پر دے اور اس کو متعلقہ آؤٹ پٹ مل جائے۔

6ماہ تک	1000	ڈرائیونگ کے لیے نااہل قرار دیے جانے کے باوجود ڈرائیونگ کرنا۔	A20
6ماہ تک	500-100	ڈرائیونگ لائسنس کے لیے درکار کاغذات کے بغیر اپلائی کرنا۔	A21
6ماہ تک	500-100	گاڑی بنانے کے متعلقہ جرم کا ارتکاب کرنا۔	A22
6ماہ تک	1000-2000	اجازت نامے کے متعلقہ جرم۔	A23
1ماہ تک	1000-3000	زیادہ بوجھ لادنا۔15% تک	A24
1ماہ تک	1000-5000	زیادہ بوجھ لادنا۔30% تک	A25

کتابیاتی نوٹ:

http://cs4schooling.com

http://nen.wikipedia.org/wiki/flowchart

مزید کوڈ جاننے کے لیے وزٹ کریں: http://nhmp.you.pk

فلو چارٹ کو مزید جامع بنانے کے لیے زیادہ سے کوڈ بڑھاتے جائیں۔ ان کے الگورتھم چارٹ پر بنائیں۔ یہ چارٹ سکول میں آویزاں کریں یا معاشرتی سطح پر ٹریفک سے متعلق جرائم اور سزاؤں کے بارے میں آگاہی فراہم کریں۔ اساتذہ کرام سکول انتظامیہ سے درخواست کریں کہ ٹریفک کا شعور بیدار کرنے کے لیے کمیونٹی میں ایک مہم چلائیں۔ اس سلسلے میں طلبہ مہم میں شرکت کر کے اپنے چارٹ دیکھا سکتے ہیں۔