# باب نمبر 5 ( سب پروگرام اور فائل ہینڈلنگ )

**سوال نمبر 1:-** سب پروگرام کیا ہوتا ہے؟ ہم اسے کیوں استعمال کرتے ہیں؟ بیسک میں ہم اسکی کتنی اقسام کو استعمال کرتے ہے

جواب :- سب پروگرام ہدایت کا سیٹ ہوتا ہے جو مخصوص کام سرانجام دیتے ہیں۔ پروگرامنگ میں بعض دفعہ ایک کام کو بار بار سرانجام دینے کی ضرورت پڑتی ہے مثلاً ہر طالب علم کے نتیجہ کو پرنٹ کرنا۔ اس صورت حال میں سلسلہ وار ہدایات لکھنی پڑے گی تاکہ یہ کام سرانجام دیا جاسکے اس کیلئے اب دو طریقے استعمال کیے جاسکتے ہیں ایک تو یہ کہ ان ہدایات کے سیٹ کو مختلف جگہوں پر لکھیں تو اس سے پروگرام سائز میں بڑا اور پیچیدہ بن جائے گا۔ دوسرا طریقہ یہ کہ ہدایات کے سیٹ کو صرف ایک بار لکھ کر بار بار پروگرام کے کسی بھی حصے سے عمل درآمد کرایا جاسکتا ہے جب بھی ضرورت ہو۔ اس دوسرے طریقے کو سب پروگرامنگ یعنی بڑے پروگرام کو چھوٹے چھوٹے حصوں میں تقسیم کرنا کہتے ہیں۔ سب پروگرام کا اپنا نام ہوتا ہے اور اس سے بڑے پروگرام کیطرح کا ملایا جاتا ہے اور پروگرام کے کسی بھی حصے میں لکھا جاسکتا ہے اور عمل درآمد کرایا جاسکتا ہے اصل میں یہ سب پروگرام بڑے پروگرام (Main Program) کا حصہ ہوتا ہے اور یہ بڑے پروگرام کے اندر رکھا جاتا ہے تصویری شکل میں اسطرح دکھایا جاسکتا ہے

| Main Program<br>{ Body of Main Program } |
|---|
| |
| Sub Program<br>{ Body of Sub Program } |

جب سب پروگرام سے عمل درآمد کرانا ہو تو کنٹرول سب پروگرام کو منتقل ہو جاتا ہے اور کام مکمل ہونے پر کنٹرول واپس مین پروگرام کو آجاتا ہے۔

بیسک زبان میں ہمارے پاس دو اقسام کے سب پروگرام ہیں

(a) فنکشن سب پروگرام (Function Sub Program) اور (b) سب روٹین سب پروگرام

a) فنکشن سب پروگرام :- یہ سب پروگرام کی ایک قسم ہے جو الجبری فنکشن کیطرح کام کرتا ہے فنکشن پروگرام دو طرح کے ہوتے ہیں (۱) بلٹن یا انٹرنزک فنکشن (یہ بیسک زبان کا حصہ ہوتا ہے مثلاً الجبری فنکشن وغیرہ) اور (۲) خود ساختہ فنکشن (یا ایسے فنکشن ہوتے ہیں جو استعمال کنندہ خود اپنی آسانی کیلئے بناتا ہے) مثالیں۔ کسی رقم کا مربع معلوم کرنا، کسی رقم کا لاگرتم معلوم کرنا وغیرہ وغیرہ

b) سب روٹین پروگرام:- یہ چھوٹے پروگرام ہوتے ہیں اور سارے کام جو بڑا پروگرام کرسکتا ہے اسکے ذریعے کرایا جاتا ہے اسکے ذریعے پیچیدہ مسائل کا حل بھی تلاش کرایا جاسکتا ہے۔

مثالیں :- میٹرکس کا حل معلوم کرنا، کوادریٹک مساوات کا روٹ معلوم کرنا وغیرہ وغیرہ اسکے GOSUB اور ON/GOSUB کی ہدایات استعمال کی جاتی ہے۔

**سوال نمبر 2:-** انٹرنزک فنکشنز (Intrinsic Functions ) کا کیا مقصد ہے؟ ہم اسے کیوں استعمال کرتے ہیں۔ موزوں مثالیں دیکر انٹرنزک فنکشنز کی وضاحت کریں؟

جواب :- داخلی فنکشنز (انٹرنزک فنکشنز) سسٹم یا کمپرو یا لائبریری فنکشنز وہ افعال ہوتے ہیں جو بیسک زبان کا حصہ ہوتے ہیں مخصوص کام سرانجام دیتے ہیں۔ یہ فنکشنز بہت سے بنیادی کاموں کو سرانجام دیتے ہیں مثلاً جذر معلوم کرنا، سائن، کوسائن یا کسی زاویے کا ٹینجنٹ معلوم کرنا وغیرہ وغیرہ۔ جیسا کہ یہ تمام کام استعمال ہونے والے ہیں یہی وجہ ہے کہ ان جیسے تمام فنکشنز کو بیسک کا حصہ بنا دیا گیا ہے اور جب انکی ضرورت پڑتی ہے تو صرف انکا عمل درآمد کیلئے لکھنا پڑتا ہے اور استعمال کنندہ کو تعریف کی ضرورت نہیں پڑتی۔ ان فنکشنز کو دو بڑے گروہوں میں تقسیم کیا گیا ہے ۱) نیومیرک فنکشنز (۲) سٹرنگ فنکشنز

**(1):- نیومیرک فنکشنز (Numeric Functions)** :- بیسک زبان میں بڑی تعداد میں بنے ہوئے (builtiu function) فنکشنز موجود ہیں جو حسابی مسائل (کے حل) کیلئے ہوتے ہیں یہ تمام فنکشنز عدد میں واپس جواب دیتے ہیں چند مندرجہ ذیل ہیں۔

(a) ٹریگنومیٹرک فنکشنز:- ان میں مندرجہ ذیل فنکشنز شامل ہے جیسا کہ SIN(x), COS(x), TAN(x), ATN(x) ان تمام فنکشنز میں (x) کی قیمت ہمیشہ ریڈین میں ہونا چاہیے

مثال:-

```
10 CLS
20 PRINT "ANGLE","SIN","COS","TAN"
30 LET F=3.14159/180
40 FOR 1=1TO 5
50 READ N
60 PRINT N.SIN(F*N),COS(F*N),TAN(F*N)
```

AwarenessStream.com

```
70 NEXT 1
80 DATA 0,1,45,60,90
90 END
```

جواب کے طور پر ٹرگنومیٹرک فنکشنز کی قیمتوں کا ٹیبل بن جائے گا۔

b) سادہ حسابی فنکشنز:۔ ان میں مندرجہ ذیل فنکشنز شامل ہیں جیسا کہ CINT(x),FIX(x),RND(x),SGN(x),INT(x),ABS(x),EXP(x),LOG(x),SQR(x), وغیرہ وغیرہ

مثال۔

```
SQR(9)      3,  ABS(-10)    10 ,  SNG(9)
INT(+9.0)   9,  FIX(-3.7)   -3,   CINT(50.25)   50
LOG(1)   0,  EXP(1)   2.71828
```

(2):- سٹرنگ فنکشنز (String Functions) :- غیر عددی فنکشنز بنے بنائے عددی فنکشنز کے علاوہ بیسک زبان میں بنے بنائے غیر عددی سٹرنگ فنکشنز بھی موجود ہیں جو غیر عددی ڈیٹا پر عمل درآمد کرتے ہیں۔ جو مندرجہ ذیل ہیں

MID$(String,[m]n), RIGHT$(String,n), LEFT$(String,n)

مثال:۔

```
10 CLS
20 PRINT LEFT$(Y$,2)
30 PRINT RIGHT$(Y$,8)
40 PRINT MID$(Y$,3,5)
50 END
```

awarenessStream.com

اس پروگرام کو چلانے کے بعد مندرجہ ذیل نتیجہ نظر آئے گا۔

```
MY
COMPUTER
COMPU
```

دوسرے غیر عددی فنکشنز مندرجہ ذیل ہیں۔ 1:- SPACES$(n) یہ خالی جگہوں کی سٹرنگ پیدا کرتا ہے
2:- STR$(x) ۔ یہ عددی قیمت کو غیر عددی قیمت میں تبدیل کرتا ہے 3:- CHR$(n) :- یہ 'n' کے طور پر دیئے گئے اسکی کوڈ کا حرف معلوم کرتا ہے جبکہ n کی قیمت 0 سے لیکر 255 تک ہوتی ہے۔

مثال:۔

```
10 LET x =1001
20 LET y =2002
30 LET z=3333
40 PRINT STR$(x).+STR$(y).+STR$(z)
50 END
```

جواب: 1001 2002 3333

جواب:

```
10 PRINT CHR$(67)..........C
10 PRINT " Pakistan" SPACES$(10)"zaindabad"
```

اس طرح پاکستان اور زندہ باد کے درمیان 10 خالی جگہیں پیدا ہو جائیں گی۔

## سوال نمبر 3:- آپ خود ساختہ فنکشنز (User Defined Functions) کے بارے میں کیا جانتے ہیں؟ ہمیں اسکی استعمال کی کب ضرورت پڑتی ہے۔

جواب:۔ جیسا کہ نام سے ظاہر ہے کہ یہ بیسک زبان کا حصہ نہیں ہیں بلکہ پروگرامر ان کو اس وقت بناتے ہیں جب بنا بنایا موزوں فنکشن نہ ملے۔ جب اس کو بنا دیا جاتا ہے تو یہ بنے بنائے فنکشن کی طرح کام کرتا ہے مثلاً کسی دائرے کا رقبہ معلوم کرنے کیلئے خود ساختہ فنکشن بنانا پڑتا ہے۔ ان فنکشن کو DEF FN کی ہدایات کی مدد سے بیان کیا جاتا ہے اسکا ترتیب نحوی مندرجہ ذیل ہے

LINE# DEF FN name (argument)=Expression

جہاں DEF کے ذریعے فنکشن بنانے کے بعد فنکشن کا نام اور جسکے پہلے دو حرف FN ہوتے ہیں اسکے بعد بریکٹ میں ایک یا ایک سے زیادہ ویری ایبلز لکھے جاتے ہیں۔ جسے آرگومنٹ کہتے ہیں اگر آرگومنٹ کی تعداد ایک یا ایک سے زیادہ ہو تو یہ ایک دوسرے سے کوما کی مدد سے جدا کیے جاتے ہیں آرگومنٹ کے بعد وہ فارمولا یا ہدایات لکھی جاتی ہے جسے فنکشن کے ساتھ منسوب کرانا ہو۔ ایک فنکشن کیساتھ ایک ہی ہدایت لکھی جا سکتی ہے۔ یاد رکھیں کہ بریکٹ کے اندر جو ویری ایبلز یا آرگومنٹس لکھے جاتے ہیں وہ صرف فنکشن کیلئے استعمال کیے ہوتے ہیں۔ پروگرام کے دیگر ویری ایبلز کے ساتھ انکا کوئی تعلق نہیں ہوتا اور نہ ہی ان پر یہ اثر انداز ہو سکتے ہیں فنکشن کو استعمال کرنے سے پہلے برابر آرگومنٹس کیلئے انکی قدر مہیا کی جاتی ہے یعنی آرگومنٹس کی تعداد کے مطابق بالترتیب قدریں مہیا کرنی چاہیے

خودساختہ فنکشن عددی یا غیر عددی ہوسکتے ہیں لہذا ان کے مطابق فنکشن کے نام اور آرگومنٹس منتخب کرنی چاہیے ورنہ اس پر کمپیوٹر ایک غلطی کی نشاندہی کریگا۔ جب ایک دفعہ DEF FN کی ہدایت سے پروگرام میں خودساختہ فنکشن بن جاتے تو اسکے بعد یہ فنکشن پروگرام میں کہیں بھی استعمال کیا جاسکتا ہے جب بھی ضرورت پڑتی ہے۔

مثال۔

```
10 CLS
20  PI=3.14159
30  DEF  FN  CIRCUMFERENCE(R)=2*PI*R
40  INPUT "Enter the radius of the circle;"RADIUS"
50  PRINT "the circumference of circle is;"FN circumference(RADIUS)
60 END
```

اس پروگرام کو چلانے پر مندرجہ ذیل جواب دیگا۔

```
Enter the radius of ?  1.2
the circumference of circle is  75.39816
Ok
```

## سوال نمبر 4:۔ سب روٹین سب پروگرام سے کیا مراد ہے؟ یہ کس طرح ماڈیولر پروگرامنگ میں معاون ہے؟ جب سب روٹین کو استعمال کرنا ہو تو کیا احتیاط لازمی طور پر کرنی چاہیے۔

جواب :۔ سب روٹین سب فنکشن پروگرام کیطرح سب پروگرام کی ایک قسم ہے دراصل سب روٹین بیسک زبان کی ہدایات کا ایک سیٹ ہوتا ہے۔ جو ایک بار لکھے جاتے ہیں اور پھر انکو جتنی بار چاہے استعمال کیا جاسکتا ہے ایک مخصوص کام سرانجام دینے کیلئے۔ مثلاً ایک عدد کے فیکٹوریل کو معلوم کرنے کیلئے ایک سب روٹین بنا کر جتنی دفعہ چاہے ضرورت ہو پروگرام کے کسی بھی حصے میں استعمال کیا جاسکتا ہے سب روٹین کوئی نام نہیں دیا جاتا ہے کیونکہ یہ ہدایات کا ایک سیٹ ہوتے جو ایک یونٹ کی کام کرتے ہیں تاہم ایک سب روٹین GOSUB کی ہدایات سے شروع ہوتا اور RETURN ختم ہے جاتا ہے یعنی GOSUB کی ہدایات کنٹرول کو سب روٹین پر منتقل کرنے کیلئے کیا جاتا ہے اس کا ترکیب نحوی یہ ہے INE# GOSUB LINE#

مثلاً

```
40  GOSUB 60
```

جبکہ RETURN کنٹرول کو واپس GOSUB ہدایت کے نیچے والی لائن پر منتقل کرو یتا ہے مثلاً

```
90   RETURN
```

نوٹ۔ یہ سب روٹین کے اختتام کو ظاہر کرتا ہے۔

اب کنٹرول لائن نمبر پر 50 منتقل ہوجائے گا کیونکہ لائن نمبر 40 پر GOSUB کی ہدایات ہے

مثال:۔

```
10 GOSUB 60
20 PRINT "High  School  No  1"
30 PRINT "Bannu".
40 END
50 PRINT "I Read Computre Subject"
60   PRINT " Class 10TH   SECTION   C"
70 RETURN
```

awarenessStream.com

جب پروگرام کو چلایا جائے تو لائن نمبر 10 پر GOSUB کی ہدایات کنٹرول کو لائن نمبر 50 پر منتقل کرویگا۔ لائن نمبر 50 اور 60 پر عمل کرنے کیبعد لائن نمبر 70 پر RETURN کی ہدایات کنٹرول کو واپس GOSUB کی نیچے لائن 20 پر منتقل کرو یگی۔

آؤٹ پٹ:۔

```
I Read Computre Subject
Class 10TH   SECTION   C
High  School  No  1
Bannu.
```

سب روٹین بڑے پروگرام کو چھوٹے حصوں (ماڈیولر) میں تقسیم کرنے کا موثر طریقہ فراہم کرتا ہے۔ اسکا فائدہ یہ ہے کہ ہدایت کو بار بار لکھنے کی ضرورت نہیں ہوتی۔ پروگرام کے لاجک کو سمجھنے میں بہت آسانی ہوتی ہے پروگرام میں تبدیلی کرنا اور غلطیوں کو دور کرنا اور کو پڑھنے میں کافی آسانی ہوجاتی ہے اسی وجہ سب روٹین کو عام طور پر استعمال کیا جاتا ہے۔

## سوال نمبر 5:۔ مندرجہ ذیل فنکشنز کا کیا مقصد ہے؟

(a COS(x) ,SIN(x)   (b ATN(x), TAN(x)   (c RIGHT$() ,  LEFT$()

(d DATE$(), TIME$()   (e VAL(), LEN()

جواب:۔ (a) SIN(x): ۔یہ ایک ٹرگنومیٹرک فنکشنز ہے جو کسی عدد کا سائن معلوم کرنے کیلئے استعمال کیا جاتا ہے مثلاً، =X تو ( SIN کا جواب

COS(x) ۔یہ ایک ٹرگنومیٹرک فنکشن ہے جو کسی عدد کا سائن معلوم کرنے کیلئے استعمال کیا جاتا ہے مثلاً، =X تو ( )COS کا جواب

(b) TAN(x): ۔یہ ایک فنکشن ہے جو کسی عدد کا ٹینجنٹ معلوم کرنے کیلئے استعمال کیا جاتا ہے مثلاً =X تو )TAN کا جواب

ATN(x) ۔یہ ایک فنکشن ہے جو کسی عدد کا آرکٹینجنٹ معلوم کرنے کیلئے استعمال کیا جاتا ہے یعنی ٹینجنٹ کا معکوس۔ مثلاً =X تو ( ATN کا جواب

(c) LEFT$() ۔یہ ایک غیر عددی فنکشن ہے جو کسی سٹرنگ (غیر عدد) بائیں طرف پر 'n' حروف کو چن کر پرنٹ کرنے کیلئے استعمال کیا جاتا ہے اگر 'n' کی قیمت سٹرنگ میں موجود حروف

کی تعداد سے زیادہ ہو تو مکمل پرنٹ ہو جائے گا۔

مثلاً

awarenessStream.com

```
10 LET Y$= "Rahamt Ullah Khan"
20 PRINT LEFT$(Y$,6)
30 PRINT LEFT$(Y$,12)
```

آؤٹ پٹ۔

AMJAD
AMJAD Ullah

کیونکہ n کی قیمت سٹرنگ میں موجود حروف سے زیادہ ہے

RIGHT$() :- یہ ایک غیر عددی فنکشن ہے جو کسی غیر عدد (سٹرنگ) کے دائیں جانب n حروف کو چن کر پرنٹ کرتا ہے اور اگر n کی قیمت سٹرنگ میں موجود حروف کی تعداد سے زیادہ ہو تو مکمل سٹرنگ کو پرنٹ کرے گا۔ مثلاً

```
10 LETY$="Bannu Gul"
20 PRINT RIGHT$(Y$,2)
30 PRINT RIGHT$(Y$,3)
```

آؤٹ پٹ۔

ul
Gul

(d) TIME$()۔ یہ ایک بنا بنایا بیسک زبان کا فنکشن ہے جس کا تعلق سسٹم کے کلاک ٹائم سے ہے یہ فنکشن میں موجود ٹائم کے ٹائم شو کرنے نیا سیٹ کرنے کیلئے استعمال کیا جاتا ہے یہ موجود سسٹم ٹائم کو HH:MM:SS کے فارم میں یعنی HH گھنٹوں (۰۰ سے ۵۹ تک)، MM منٹوں (۰۰ سے ۵۹) اور SS سیکنڈز (۰۰ سے ۵۹) کی صورت میں پیش کرتا ہے۔

مثلاً۔

```
10 PRINT TIME$
```

نیا ٹائم سیٹ کرنے کیلئے یہ ہدایات استعمال ہوگی۔

```
TIME$ = "HH:MM:SS"
10 TIME$ ="16:00:00"
```

مطلب 4:00 PM بجے

DATE$()۔ یہ ایک داخلی بنا بنایا بیسک زبان کا فنکشن ہے۔ جس کا تعلق سسٹم کے ڈیٹ سے ہے یہ فنکشن سسٹم کے موجودہ تاریخ کو دیکھنے یا سیٹ کرنے کیلئے استعمال کیا جاتا ہے یہ تاریخ کو DD:MM:YYYY کے صورت میں پیش کرتا ہے (یعنی DD دنوں، MM مہینوں اور YYYY سال کی صورت میں) تاریخ ہمیشہ لمبا اور ڈیش (-) یا سلیش (/) کے ذریعے جدا ہوتا ہے۔ سسٹم ڈیٹ کو دیکھنے کیلئے LINE# DATE$ کی ہدایات استعمال کی جاتی ہے مثلاً۔ 10 DATE$

جبکہ سسٹم ڈیٹ کو سیٹ کرنے کیلئے LINE# DATE$="NEW DATE " کی ہدایات استعمال کی جاتی ہے

مثلاً۔

```
10 DATE$="03-03-1989"
10 DATE$="15/12/2005
```

آؤٹ پٹ۔

03-03-1989
15/12/2005

یا

(e) LEN()۔ LEN(x$) ایک فنکشن ہے جو ایک غیر عدد (سٹرنگ) میں موجود تمام حروف کی تعداد کو پرنٹ کرنے کیلئے استعمال کیا جاتا ہے۔ X$ وہ سٹرنگ ہے جس کی حروف کی تعداد کو پرنٹ کرتا ہے اگر سٹرنگ میں کوئی حروف نہ ہو تو LEN فنکشن 0 کی صورت میں قیمت جواب دیتا ہے۔ یعنی سٹرنگ کی لمبائی معلوم کرنے کیلئے استعمال ہوتا ہے۔ مثلاً۔ LINE# LEN(X$)

LEN("AMJAD ULLAH KHAN")

آؤٹ پٹ 17 ہوگا بشمول خالی جگہ کے۔ LEN (" ") آؤٹ پٹ 0 ہوگا۔

VAL() :- VAL(X$) فنکشن کسی سٹرنگ (جو عددی صورت میں ہو) کو عددی قیمت کی صورت میں تبدیل کرنے کیلئے استعمال ہوتا ہے بشرطیکہ سٹرنگ ہندسوں پر مشتمل ڈبل کوٹیشن ("") میں بند ہو۔ اگر سٹرنگ غیر عددی ہو تو جواب 0 کی صورت میں آئے گا۔

مثال۔

```
10 A$="03339741577"
20 B=VAL(A$)
```

آؤٹ پٹ

03339741577

لائن نمبر 10 میں A$ کی قدر قیمت "03339741577" مقرر کی گئی ہے جو کہ ایک سٹرنگ ہے۔ بظاہر تو یہ ایک نمبر نظر آتا ہے لیکن کمپیوٹر اسے نمبر نہیں سمجھتا کیونکہ اس کا نام سٹرنگ کا نام ہے اور یہ انورٹڈ کوموں (" ") میں بند ہے۔ لائن نمبر 20 میں "03339741577" سٹرنگ سے نمبر بن جاتا ہے اب اس کا نام B ہے۔

## سوال نمبر 6:۔ فائل سے کیا مراد ہے۔ فائل کی تنظیم کے کیا معنی ہے؟ وضاحت کریں۔

جواب:۔ سیکنڈری سٹوریج آلات (ٹیپ، ڈسک وغیرہ) میں وہ جہاں پر ڈیٹا کو مستقل طور پر محفوظ کیا جاتا ہے فائل کہلاتا ہے۔ اس جگہ کو فائل کا نام دیکر جب چاہے ڈیٹا کو دیکھا جاتا ہے عموماً فائل کی

awarenessStream.com (object) کے متعلق ریکارڈ کے مجموعے جو سیکنڈری سٹوریج آلات پر محفوظ کیا جاتا ہے کو فائل کی تنظیم کہلاتی ہے

بیسک زبان میں فائل کے اندر ریکارڈ کو مندرجہ ذیل طریقوں سے منظم کیا جاتا ہے

(۱) ترتیب وار فائل (۲) براہ راست رسائی والی فائل ان کی وضاحت درج ذیل ہے

(۱) ترتیب وار فائل:۔ یہ نہایت سادہ فائل کی تنظیم ہے جسمیں ڈیٹا کو ترتیب وار طریقے سے فائل میں محفوظ اور دیکھا جاتا ہے اس ترتیب میں ڈیٹا یا معلومات کو دیکھے بغیر ایک ریکارڈ کے بعد دوسرے اور دوسرے کے بعد تیسرا ریکارڈ محفوظ کر دیا جاتا ہے اور اسی طرح ایک خاص ترتیب میں ڈیٹا کو دیکھا جا سکتا ہے۔ مثلاً 100 ریکارڈ کو محفوظ کرنا ہے تو ایک سے شروع کرنے کے بعد 100 تک کے ریکارڈ کو محفوظ کرینگے اب آپ ریکارڈ نمبر 50 دیکھنا ہے تو پہلے 49 ریکارڈ کو دیکھنے کے بعد 50 نمبر کا ریکارڈ دیکھ سکینگے جس کی وجہ سے وقت کافی ضائع ہوتا ہے۔ فائل کے اندر ریکارڈز کو اوپر نیچے کی جانب یا نیچے درکے جاتے ہیں ان میں سے کوئی ترتیب دینے کسی فیلڈ کا انتخاب کیا جاتا ہے جسے کی فیلڈ کہا جاتا ہے اور پھر اسی فیلڈ کے مندرجات کے لحاظ سے پوری فیل کی ترتیب بدل دی جاتی ہے کی (Key) ناموں یا ارقام والی فیلڈ یعنی نومیرک فیلڈ یا دونوں کا انتخاب کیا جا سکتا ہے۔ مثلاً

| CustomerID | Cus: Name | Customer Age |
|---|---|---|
| 0301 | AMJAD | 26 |
| 0304 | FARMAN | 18 |
| 3032 | AWAIS | 20 |
| 0333 | FIDA | 30 |
| 974 | SAQIB | 21 |
| 2747 | FAHAD | 15 |

Key Field

| Sr. no. | Name | Age |
|---|---|---|
| 120 | AMJAD | 26 |
| 130 | FARMAN | 10 |
| 110 | AWAIS | 40 |
| 100 | FIDA | 70 |
| 125 | SAQIB | 50 |
| 725 | FAHAD | 30 |

Key Field

نقصانات:۔ ترتیب وار فائل میں مزید ریکارڈ کو داخل کرنا یا کسی ریکارڈ کو فائل میں سے مٹانا بہت مشکل ہوتا ہے اور بعض صورتوں میں ناممکن ہے لہذا ایسے ماحول میں جہاں معلومات کی فوری ضرورت پڑتی ہے اس قسم کی فائل ناموزوں خیال کی جاتی ہے۔

(2) براہ راست رسائی والی فائل (Random or Direct Access Files):۔ براہ راست رسائی والی فائل میں ریکارڈ براہ راست محفوظ اور ان کی براہ راست دیکھی جا سکتی ہے اسلئے اس براہ راست رسائی والی فائل کی تنظیم کہا جاتا ہے۔ یعنی کسی مخصوص ریکارڈ کو تمام ریکارڈ تک رسائی کے بغیر پہنچا جا سکتا ہے۔ جسکی وجہ سے کسی مخصوص فائل کو تلاش کرنے میں بہت کم وقت لگتا ہے اور یہی اسکی ترتیب وار فائل کی تنظیم کے مقابلے میں فائدہ کی بات ہے اسی طرح ریکارڈ کو بے ترتیبی یا براہ راست فائل میں کسی بھی جگہ پر محفوظ یا جمع کیا جا سکتا ہے اس فائل کی تنظیم کی بنیاد ریکارڈ کی (Key) اور ڈسک میں ریکارڈ کی جگہ کے مابین تعلق پر رکھی گئی ہے۔ یعنی ریکارڈ کی (Key) فائل میں ریکارڈ کی اصل جگہ کو ظاہر کرتی ہے۔

استعمال:۔ اس تنظیم کو ایسے فائل بنانے میں استعمال کیا جاتا ہے جس میں انفرادی ریکارڈ کو پراسیس کرنا ہو۔ مثلاً ٹکٹ والی ایجنسیوں اور ہوائی نشست کی تخصیص کا انتظام وغیرہ۔

نقصانات:۔ اس تنظیم کے فائل بنانے کا پروگرام مشکل ہوتا ہے۔ سب سے مشکل فائل میں ڈیٹا کی بناوٹ کی طریقہ طلاق ہیشنگ (Hashing) کہلاتی ہے جو براہ راست رسائی والی فائل کی تنظیم میں استعمال ہوتی ہے اور یہی اس کی بڑی خامی ہے۔

## سوال نمبر 7:۔ مندرجہ ذیل کے متعلق آپ کیا جانتے ہیں۔

a) اوپن کی ہدایت OPEN Statement — b) کلوز کی ہدایت CLOSE Statement

c) رائٹ # کی ہدایت WRITE# Statement — d) پرنٹ # کی ہدایت PRINT# Statement

e) ان پٹ # کی ہدایت INPUT# Statement — a) اوپن کی ہدایت (OPEN Statement)

جواب:۔ a) اوپن کی ہدایت OPEN Statement:۔ بیسک زبان کے پروگرام میں اوپن (OPEN) کی ہدایت اوپریٹنگ سسٹم کو I/O ان پٹ / آؤٹ پٹ کے پر (مثلاً ڈسک) ایک فائل کھولنے کی ہدایت دیتی ہے۔ فائل کو دو مختلف مقاصد کیلئے کھولا جا سکتا ہے۔ یعنی ان پٹ کیلئے یا آؤٹ پٹ کیلئے۔ OPEN ہدایت کی ترکیب نحوی ذیل ہے:

```
LINE# OPEN "MODE",[#] FILE number,"File name",[Record len]
```

جہاں بڑی بریکٹ میں ایکسپریشن اختیاری ہے۔

MODE:۔ موڈ دراصل ایک سٹرنگ ایکسپریشن ہے یہ فائل تک رسائی کی قسم کو ظاہر کرتا ہے کوئی بھی فائل مندرجہ ذیل چار موڈ (انداز) میں کھولا جا سکتا ہے

"O" موڈ:۔ O موڈ ترتیب وار آؤٹ پٹ کو ظاہر کرتا ہے یعنی یہ موڈ (MODE) ترتیب وار فائل میں استعمال ہوتا ہے اگر فائل میں کچھ لکھنا مقصود ہو۔ جب اس موڈ کیساتھ فائل کو کھولا جاتا ہے تو پہلے سے "I" موجود ڈیٹا ضائع ہو جاتی ہے

"I" موڈ:۔ یہ موڈ ترتیب وار ان پٹ موڈ کو ظاہر کرتی ہے یعنی یہ موڈ ترتیب وار فائل سے ڈیٹا کو پڑھنے کیلئے استعمال کی جاتی ہے۔ جب بھی فائل اس موڈ کیساتھ کھولا جاتا ہے تو پوائنٹر

(Pointer) خودبخود فائل میں پہلے ریکارڈ پر چلا جاتا ہے۔

"A" موڈ۔ یہ ترتیب وار اپینڈ موڈ کو ظاہر کرتا ہے جب بھی فائل اس موڈ کے ساتھ کھولا جاتا ہے تو خودبخود فائل پوائنٹر فائل کے آخر پر چلا جاتا ہے اس کی وجہ سے پہلے سے موجود ڈیٹا ضائع نہیں ہوتا۔

"R" موڈ۔ یہ براہ راست (بے ترتیب) ان پٹ / آؤٹ پٹ موڈ کو ظاہر کرتا ہے براہ راست براہ راست رسائی والی فائل میں اس موڈ کے ذریعے فائل سے ڈیٹا کو پڑھا اور لکھا جاتا ہے

۔

(#) فائل نمبر۔ فائل نمبر ہر فائل کی یکتا شناخت کیلئے استعمال کیا جاتا ہے بجائے فائل کا نام وغیرہ کے ہر وقت جب فائل تک رسائی کرتی ہے۔ جب تک فائل کھولا ہوا اس وقت تک یہ (فائل) نمبر فائل سے منسلک رہتا ہے۔ ایک جیسا نمبر دو مختلف فائلوں کو نہیں دینا چاہیے۔ یہ لازمی طور پر عدد ہونا چاہیے اور ڈسک کیلئے۔۔۔۔۔۔ مثلاً #1 , #2 , #3۔

فائل کا نام (File Namae)۔ یہ فائل کا حقیقی نام ہے جسے یہ ڈسک پر محفوظ ہوتا ہے یہ ایک غیر عددی (سٹرنگ) ہوتا ہے اور لازماً ڈبل کوٹز میں بند ہوتا ہے یعنی فائل کا نام آٹھ حروف اور ایکسٹینشن تین حروف پر مشتمل ہو سکتے ہیں۔ فائل کا نام اور ایکسٹینشن ڈاٹ (.) کے ذریعے جڑے ہوتے ہیں۔ مثلاً "FARMAN.DAT"۔

ریکارڈ لینتھ (Record len) یہ ترتیب وار فائل میں اختیاری طور پر استعمال ہوتا ہے یہ براہ راست رسائی والی فائل میں ریکارڈ کی لمبائی کو سیٹ کرنے والے کیلئے استعمال ہوتا ہے ریکارڈز کی لمبائی 1 تا 32767 تک ہو سکتی ہے۔ اگر قیمت سیٹ نہ ہوئی گئی ہو تو اس کی اپنی خودساختہ قیمت 128 استعمال ہوگی

مثالیں:۔

```
10 OPEN "O" , # 1 ,"PAYROLL"
30 OPEN "I" , # 3 , "STUDENT"
50 OPEN "A" ,# 2 , "ACCOUNT"
70 OPEN "R" ,# 1, " DATA",100
```

**(ii) :- کلوز کی ہدایت (COLSE)۔** کلوز کی ہدایت فائل پر پروسیسنگ ختم کرنے کیلئے استعمال کیا جاتا ہے۔ کیونکہ ہر کھلی ہوئی فائل کو بند کرنا پڑتا ہے اس سے پہلے کہ پروگرام ختم ہو۔

اس کی ترتیب نحوی یوں ہے۔ LINE# CLOSE [#] file number, [#] File number,....

یہ ہدایت ایک کی لفظ CLOSE(Key word) اور اس کے بعد فائل نمبر لکھے بعد دیگر کومہ کے ذریعے ایک دوسرے سے جدا ہوتے ہیں۔ تمام فائلوں کو ایک کلوز کی ہدایات کے ذریعے یا ہر ایک کو الگ الگ کلوز کی ہدایات کے ذریعے بند کئے جاتے ہیں۔ اگر کلوز کی ہدایات کیساتھ اختیاری فائل نمبر لکھا جائے تو صرف وہی فائل بند ہوگی لیکن اگر کلوز کی ہدایت کیساتھ فائل نمبر نہیں لکھا جائے تو تمام کھلے ہوئے فائلز کلوز کی ہدایت بند کرے گا۔

مثالیں:۔

```
200 CLOSE #1
100 CLOSE #
```

**(iii) :- رائٹ (لکھنے) کی ہدایات (WRITE # Statement)۔** ترتیب وار فائل میں ڈیٹا کو لکھنے کیلئے (WRITE#) یا (PRINT#) کی ہدایت استعمال کی جاتی ہے۔ ان دونوں ہدایات کا مقصد ایک ہے لیکن آؤٹ پٹ میں فرق ہے۔ WRITE# ہدایت کی ترتیب نحوی ذیل ہے:

Line#WRITE# File Name,Exprssion,Expression,..................

جہاں فائل نمبر اس کا نمبر ہے جو (OPEN#) کی ذریعے کھولا گیا ہے۔ ایکسپریشن سٹرنگ (غیر عددی) یا نیومیرک (عددی) ایکسپریشن ہو سکتا ہے اور یہ ایک دوسرے سے کومہ (,) کے ذریعے جدا کیا جاتے ہیں۔ مثلاً:۔

```
10 OPEN"O",#1,"DATAFILE"
20 N$="AMJAD"
30 A$="khanpur"
40 Ph.No=613666
50 WRITE#1,N$,A$,Ph.No
60 CLOSE
```

awarenessStream.com

اس طرح فائل میں مندرجہ ذیل ڈیٹا محفوظ ہو جائے گا

"AMJAD","khanpur",613666

نوٹ:۔ WRITE# کی ہدایات تمام سٹرنگ کو ڈبل کوٹز (" ") میں بند اور کومہ (,) کے ذریعے سے ایک دوسرے کو جدا کرتا ہے۔

**(iv) :- پرنٹ کی ہدایت (PRINT#)۔** یہ ہدایات بھی فائل میں ڈیٹا لکھنے (محفوظ) کرنے کیلئے استعمال کیا جاتا ہے لیکن ڈیٹا کو کومہ کے ذریعے اور سٹرنگ کو ڈبل کوٹز (" ") میں بند نہیں کرتا ہے مثلاً۔

```
10 OPEN "O" ,#1,"DATAFILE"
```

```
20  N$="Farman"
30  A$="khanpur"
40  Ph.NO.=5574738
50  Mob.No.  03009877661
60  PRINT#  1,N#,A$,Ph.No,Mob.No
70  CLOSE
```

آؤٹ پٹ:۔ FARMAN khanpur 5574738

لیکن اگر ہم چاہیں کہ ڈیٹا کے درمیان کومہ ہو یعنی کومہ کے ذریعے ایک دوسرے سے جدا ہو تو (,) کومہ کو ڈبل کوٹز میں بند کر کے اسطرح پرنٹ کی ہدایت میں استعمال کرینگے۔ یعنی لائن نمبر 60 کی بجائے اسطرح لکھنے یعنی کومہ کو ڈبل کوٹز بند کرنے کے بعد سیمی کولن کے ذریعے جدا کریں۔

```
60  PRINT#1,"N$",,";A$,",";Ph.No,",";Mob.No,"
```

آؤٹ پٹ میں ڈیٹا کومہ کے ذریعے سے ایک دوسرے سے جدا ہونگے:۔ FARMAN,khanpur,5574738,03009877661.

**(V):- ان پٹ کی ہدایات (INPUT Statement) :۔** ترتیب وار فائل سے ریکارڈ پڑھنے کیلئے ان پٹ (INPUT#) کی ہدایت استعمال کی جاتی ہے اسکے لیے ضروری ہے کہ وہی نمبر اور ویری ایبل وہی ترتیب استعمال کی جائے جب فائل بناتے وقت استعمال کی گئی ہو۔ ترتیب وار فائل سے ریکارڈ پڑھنے کیلئے فائل کو "I" (ان پٹ) موڈ میں کھولنا چاہیے۔ اس ہدایات کی ترتیب نحوی مندرجہ ذیل ہے۔

```
Line# INPUT#  Filenumber,Variable1,Variable2,......................
```

فائل نمبر وہ نمبر ہے جسکے ذریعے فائل کو ڈیٹا پڑھنے کیلئے کھولا جاتا ہے یعنی ڈیٹا ان پٹ کیلئے ویری ایبل لسٹ میں ویری ایبل کے نام فائل میں آئٹم (Items) کو دیے جاتے ہیں موجود ہوتے ہیں۔ ویری ایبل کی قسم ڈیٹا کی قدر کے جیسے ہونی چاہیے ورنہ غلطی کا پیغام "TYPE Mismatch" سکرین پر آجائے گا اور پروگرام ہالٹ ہوجائے گا۔ یہ بھی سوالیہ نشان کو پرنٹ نہیں کرتا۔

مثلاً:۔ مندرجہ ذیل پروگرام کا حصہ ترتیب وار فائل سے ڈیٹا کو پڑھے گا اور سکرین پر اسے پرنٹ کرتا ہے

```
OPEN  "i",#I,"DATAFILE"
INPUT#I,N$,A$,Ph.No
PRINT N$,A$,Ph.No
CLOSE# I
```

آؤٹ پٹ:۔ FARMAN   khanpur   5574738

یعنی وہ ڈیٹا جو ہم نے پہلے فائل میں ان پٹ کیا پرنٹ ہوجائے گا۔

awarenessStream.com

**سوال نمبر 8:-** مندرجہ ذیل پرنٹ (Print) سٹیٹمنٹ سے کیا ظاہر ہوگا۔

| | | | |
|---|---|---|---|
| a | PRINT SGN (22, 74) | f | PRINT LOG (1) |
| b | PRINT INT (-8.8) | g | PRINT SIN ( ) |
| c | PRINT ABS (-17.53) | h | PRINT COS ( /2) |
| d | PRINT FIX (-17.53) | i | PRINT  TAN (0) |
| e | PRINT SQR (25) | j | PRINT CHR $ (80) |

جوابات :-

| | | | |
|---|---|---|---|
| a | | f | 0 |
| b | - 9 | g | 0 |
| c | 17.35 | h | 0 |
| d | -17 | i | 0 |
| e | 5 | j | P |

**سوال نمبر 9:-** تمام حروف (A تا Z اور a تا z) کے ASCII کو ڈ پرنٹ کرنے کیلئے پروگرام لکھیں۔

جواب :-

```
10  CLS
20  FOR A = 65 TO 90
30   PRINT CHR$(a),  a;  SPC(25);  CHR$  (a + 25), a + 32
```

```
40   NEXT
RUN
```

**سوال نمبر 10:- خالی جگہ پُر کریں۔**

i):- سب پروگرام کے دو اقسام ہیں فنکشن سب پروگرام اور (b) سب روٹین سب پروگرام

ii):- TAN ، SIN ، LOG نیومیرک فنکشنز کہلاتے ہیں۔

iii:- LEFT$ ، MID$ ، RIGHT$ سٹرنگ فنکشنز کہلاتے ہیں۔

iv:- LOG فنکشن قدرتی لوگارتھم کی قیمت معلوم کرنے کے لئے استعمال کیا جاتا ہے۔

v:- CHR$ فنکشن ASCII کوڈ نمبر n کے متناظر کریکٹر کو معلوم کرنے کے لئے استعمال کیا جاتا ہے۔

awarenessStream.com