**CONSTANTS**

The value or quantities never change during execution of a program are known as constants.

One constant occupies space of 4 bytes.

4 types

1 ) integer constants

2) decimal integer constant- a numeric value which have digits from 0-9.

3) octal integer constant- it can have digits from 0-7 and starts from 0. Eg. 026,037.

4) Hexa decimal integer constants- it can have digits from 0-9 and A-F and starts with ox. Eg. Ox37A

**Float/real constants**- A numeric value having decimal point

1. Decimal type -99.9
2. exponential type- 5.6\*10^5

**character constants** – A single alphabet capita or small , a single digit , any special symbol enclosed in single quotes. Eg. ‘a’

there are 356 characters in c++

ASCII Table

|  |  |
| --- | --- |
| A to Z | 65-90 |
| a to z | 97-122 |
| 0 to 9 | 48-57 |

**Strings-** a group of characters or a single character or no character enclosed in double quotes. Eg. “abc”

It is terminated with null character.

No of bytes consumed = no of strings +1 byte

**Variables**

7steps

1. understanding the problems
2. write the formula
3. Declare the variables with formula
4. Think about empty variables / boxes in memory
5. Assign a input value to the variables
6. Apply the formula
7. Show the output.

EG1. **Learning about formulas**

Area of rectangle

[\\area](file:///\\area) =l\*b

Int main()

{

Int L,B,A;

L=2;

B=3;

A=l\*b;

Printf(“%i”,A);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EG2. Learning about tab(8spaces) and changing line(L1-2)

main()

{

printf("hello\n");

printf("system\n");

printf("\t hello system");

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EG3. Learning about inserting scan(inserting data in cmd file)(L4)

int main()

{

float l,b,a;

printf("enter the length:");

scanf("%f",&l);

printf("enter the breadth:");

scanf("%f",&b);

a=l\*b;

printf("area=%f",a);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EG4. Learning to interchange numbers

int main()

{

int C,D,temp;

printf("enter C and d :");

scanf("%d %d",&C,&D);

temp=C;

C=D;

D=temp;

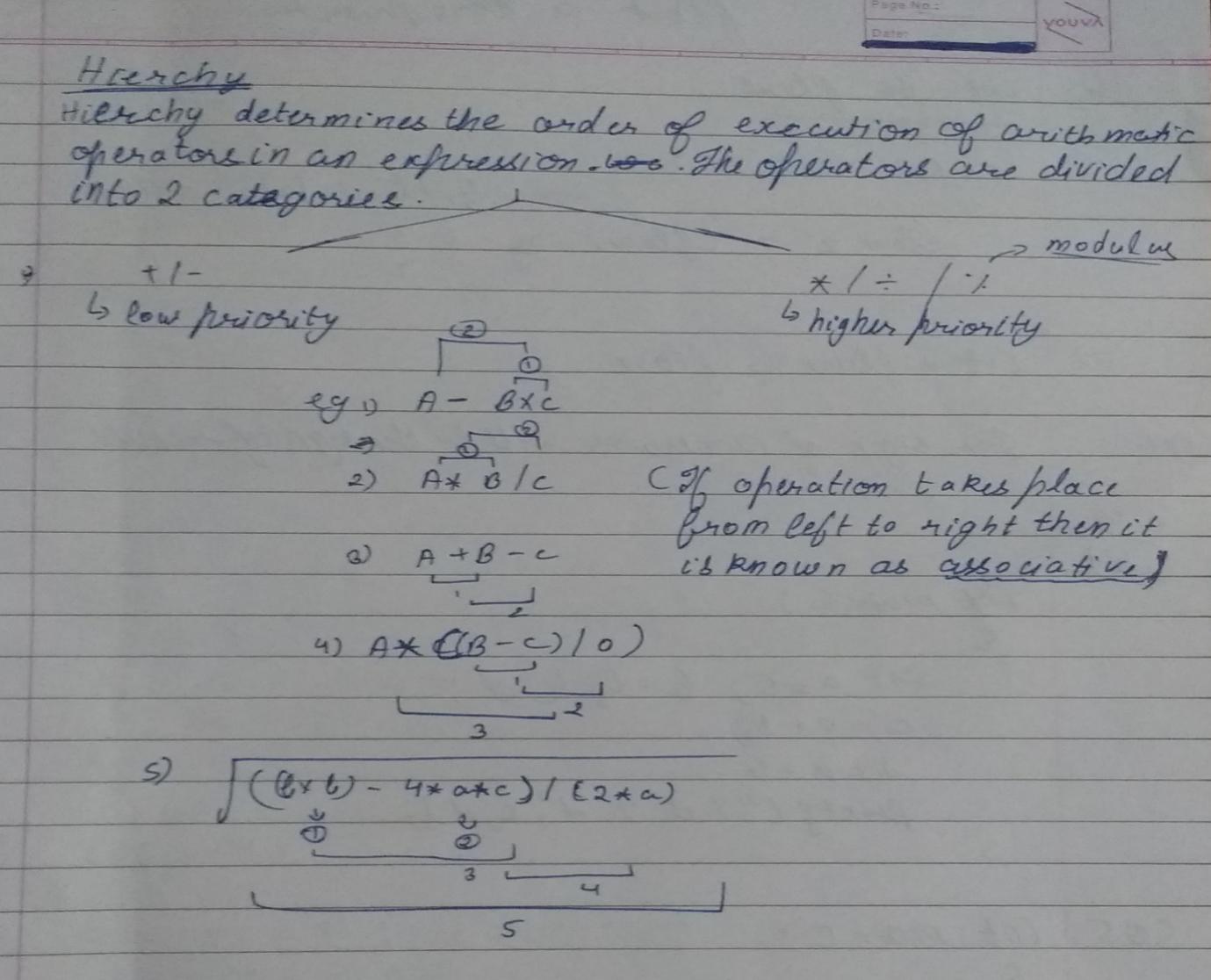
printf("C=%d \n D=%d",C,D);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Hierchy**

Hierchy determines the order of execution of arithmetic operations in an experession. The opration is divided in 2 categories.



|  |  |  |
| --- | --- | --- |
|  | Division | Modulus(reminder) |
| 1 | Int x,y;  Z=x/y=4/3=1 | Int x;  X=5%2=1 |
| 2 | Float=4/3=1.3 | Int x=8.0%3=error  (As 8.0 is float) |
| 3 | Int/float=float | Int x=5%-2=1 |
| 4 | Int/int=int | Int x=-5%-2=-1 |
| 5 | Float/int= float | -- |
| 6 | Float/float=float | -- |
| 7 | Int\*float=float | -- |

Note- the of numerator will be the sing of result.

EG5. Learning to convert int to float(integer to decimal)

int main()

{

float z;

z=5/2+(float)7/2;

printf("%f",z);

}

­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EG6. Reversing numbers and adding numbers(file hw-basic-pro10)

int main()

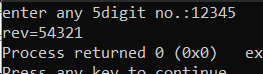
{

int n,r1,r2,r3,r4,rev,s;

printf("enter any 5digit no.:");

scanf("%d",&n); eg. 12345

r1=n%10; =5

 n=n/10; =1234

r2=n%10; =4

n=n/10; =123

r3=n%10; =3

n=n/10; =12

r4=n%10; =2

n=n/10; =1

rev=r1\*10000+r2\*1000+r3\*100+r4\*10+n;

printf("rev=%i",rev); =54321

s=r1+r2+r3+r4+n;

printf("sum=%i",s); =15

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Decision making statements**

Syntax

Int main()

{

If()

{

-------

-------

}

Else

{

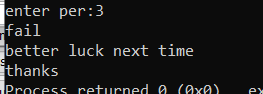
-----

-----

}

}

EG7. Learning if else(lecture=if else)

int main()

{

float per;

printf("enter per:");

scanf("%f",&per);

if (per >=50)

{

printf("pass\n");

printf("congo");

}

else

{

printf("fail\nbetter luck next time");

}

printf("\nthanks");

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* If there are multiple statements below else but without blocks then compiler will accept first statement rest will be considered as common place.
* To attach multiple statements with else we use else blocks.
* If there are multiple statements after if condition then bloack is complusary otherwise give error msg.

**Ladder if else**

It is used for making a choice among 2 or more options, eg check no -+ve,-ve,0.

Syntax

Int main()

{

If(condition)

{

}

Else

If()

{

}

Else

If ()

{

}

-

-

-

}

EG8. Electricity bill using if else (learning ladder if else)(lecture=ladder if else)

main()

{

int unit,bill,gst,gross;

printf("enter unit:");

scanf("%d",&unit);

 if(unit<=500)

bill=unit\*5;

else

if(unit<=600)

bill=unit\*6;

else

if(unit<=700)

bill=unit\*7;

gst=bill\*18/100;

gross=gst+bill;

printf("gross bill=%d",gross);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg9. +ve/-ve or zero no.s(c=notes=positive or negative)

int main()

{

int n;

printf("enter no:");

scanf("%d",&n);

if (n<0)

printf("no is -ve");

else

if (n>0)

printf("+ve");

else

printf("zero");

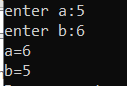
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg 10. Interchanging numbers(if else=pro1)

main()

{

 int a,b,temp;

printf("enter a:");

scanf("%d",&a);

printf("enter b:");

scanf("%d",&b);

if(a!=b)

{

temp=a;

a=b;

b=temp;

printf("a=%d\nb=%d",a,b);

}

else

printf("same");

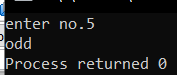
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Rational operations**

= , == , < , > , <= , >= , !=

Eg11. Finding even odd numbers (lectures=even odd)

int main()

{

int n;

printf("enter no.");

scanf("%d",&n);

if(n%2!=0)

printf("odd");

else

printf("even");

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**LOGICAL OPERATIONS**

1)(&&) and

2)(!!) or

3)(!) not

|  |  |  |  |
| --- | --- | --- | --- |
|  | C1 | C2 | Result |
| And | T | T | T |
| OR | T | F | F |
| NOT | T | NULL | F |

Eg12. Grading (lectures=grades using logical) for better method go eg15

int main()

{

int n;

printf("enter marks:");

 scanf("%d",&n);

if(n>80 && n<100)

printf(" A grade");

else if(n>70 && n<80)

printf(" B grade");

else

printf(" not eligible");

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**NESTED IF ELSE**

**Syntax**

**if(condition)**

**{**

**if(condition)**

**else**

**}**

**else**

**{**

**if(condition)**

**else**

**}**

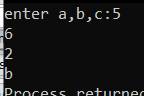
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Eg13. Finding greatest no. out of three or more(lecture=greatest number out of 3 or more) for short method go to eq 16**

**int main()**

**{**

**int a,b,c;**

** printf("enter a,b,c:");**

**scanf("%d%d%d",&a,&b,&c);**

**if(a>b)**

**{**

**if(b>c)**

**printf("a");**

**else**

**printf("c");**

**}**

**else**

**{**

**if(b>c)**

**printf("b");**

**else**

**printf("c");**

**}**

**}**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**CONDITIONAL &TURMERY OPERATIONS**

-like if-else condition for making a choice among given options.

-here we use : ,? Operators instead of if-else

syntax

(condition ? statement-1 : statement-2)

Maximum of 2 numbers

(a>b ? printf(“a”):printf(“b”));

Int max=(a>b? a:b)

Printf(%max=%d”,max);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**LADDER**

Int max=(a>b&&a>c?a:b>a&&b>c?b:c);

Printf(“%max=%d”,max);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**NESTED IF ELSE**

Int max=(a>b?a>c>a:c:b:c?b:c);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg 15. Grading by marks(hw=program=if else=pro 11)

main()

{

 int m;

printf("enter marks");

scanf("%d",&m);

(m>=80)?printf("A grade"):

(m>=70)?printf("B grade"):

(m>=60)?printf("C grade"):

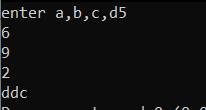
printf("fail");

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg 16 greatest no. out of three or more(hw=program=if else=pro 12)

main()

{

int a,b,c,d;

printf("enter a,b,c,d");

scanf("%d%d%d%d",&a,&b,&c,&d);

{(a>b)&&(a>c)&&(a>d)?printf("a"):printf("d");}

{(b>c)&&(b>d)?printf("b"):printf("d");}

{(c>d)?printf("c"):printf("d");}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SWITCH CASE**

* String floats are not used in switch cases.
* One can only use integer constants or characters.
* We cannot relation or logical operators in it.
* Switch, cases,default, break, these keywords are used in switch statements.

**Syntax**

Switch(choice)

{

Case1:

Break;

Case2:

Break;

Default:

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg 17. Learning to use switch case(chosing colour)(lecture= switch case-choose colour)

int main()

{

char choice;

 printf("ennter choice:");

scanf("%c",&choice);

switch(choice)

{

case 'a':

printf("red");

break;

case 'b':

printf("blue");

break;

case 'c':

printf("green");

break;

default:

printf("wrong choice");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**NESTED SWITCH CASE STATEMENTS**

Eg.18 learning switch cases in multiple steps(like phone calls coustomer care)(lecture=nested if else)

#include<stdio.h>

int main()

{

int no,a,b;

char choice;

printf("enter no(1/2/3):");

scanf("%d",&no);

switch(no)

{

case 1:

printf("enter choice(\*,/,-,+)");

scanf("%d",&choice);

switch(choice)

{

case '+':

printf("enter a&b=");

scanf("%d%d",&a,&b);

printf("sum of a&b:%d",a+b);

}

{

case'\*':

printf("enter a&b:");

scanf("%d%d",&a,&b);

printf("area of rectangle:%d",a\*b);

break;

}

{

case '-':

printf("enter a&b:");

scanf("%d%d",&a,&b);

printf("subtraction:%d",a-b);

break;

}

{

case '/':

printf("enter a&b:");

scanf("%d%d",&a,&b);

printf("division:%d",a/b);

break;

}

break;

default:

printf("wrong choice ");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**LOOPS**

* **­it is used for executing set of statements again and again until condition becomes false.**

**Syntax**

For(starting;condition;updation)

Eg19. Basic eg

int main()

{

int i;

for(i=1;i<=10;i++)

printf("%d\n",i);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg20. Factorial using loops (lecture= loops factorial)

int main()

{

 int i,fact=1,n;

printf("enter no=");

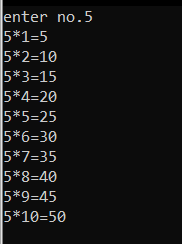
scanf("%d",&n);

for(i=1;i<=n;i++)

fact=fact\*i;

printf("%d\n",fact);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg21. Tables using loops (pro2 = hw-loops)

main()

{

int i,n;

printf("enter no.");

scanf("%d",&n);

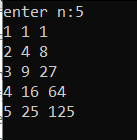
for(i=1;i<=10;i++)

printf("%d\*%d=%d\n",n,i,n\*i);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg22. Squaring and cubing using loops (hw=loops= pro3)

main()

{

int i,n;

printf("enter n:");

scanf("%d",&n);

for(i=1;i<=n;i++)

printf("%d %d %d\n",i,i\*i,i\*i\*i);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PATTERNS**

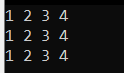
Eg23. Basic pattern(lecture=loops= pattern5)

int main()

{

int r,c;

for(r=1;r<=3;r++)

 {

for(c=1;c<=4;c++)

{

printf("%d ",c);

}

printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg24. Basic pattern(lecture=loop=pattern6)

int main()

{

 int r,c;

for(r=1;r<=3;r++)

{

for(c=8;c>=2;c=c-2)

{

printf("%d ",c);

}

printf("\n");

}

}

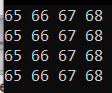
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg25. Basic pattern(lecture==loop=pattern7)

int main()

{

int r,c;

 for(r=1;r<=4;r++)

{

for(c=65;c<=68;c++)

{

printf("%d ",c);

}

printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg26. Basic pattern for charecters(lecture=loop=pro1)

int main()

{

char r,c;

for(r=1;r<=4;r++)

{

for(c='A';c<='D';c++)

{

printf(" %c",c);

}

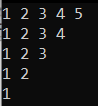
printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TRIANGULAR PATTERNS**

Eg27. Integer basic pattern(lecture=loop=triangular pattern8 loop)

int main()

{

int r,c;

for(r=1;r<=5;r++)

{

for(c=1;c<=r;c++)

{

printf("%d ",c);

}

printf("\n");

}

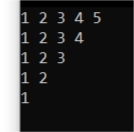
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg28. Integer basic pattern(no file)

int main()

{

 int r,c;

for(r=5;r>=1;r--)

{

for(c=1;c<=r;c++)

{

printf("%d ",c);

}

printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg.29. integer basic pattern(lecture=loop=pattern9triangular)

int main()

{

int r,c;

for(r=1;r<=4;r++)

{

for(c=r;c>=1;c--)

{

printf("%d ",c);

}

printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg30. Character basic pattern(lecture=pattern 10)

int main()

{

char r,c;

for(r='a';r<='d';r++)

{

for(c='a';c<=r;c++)

{

printf("%c ",c);

}

printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EG31. Special pattern (lecture=pattern11)

int main()

{

int i,j,no=0;

 for(i=1;i<=4;i++)

{

for(j=1;j<=i;j++)

{

no++;

printf("%d ",no);

}

printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

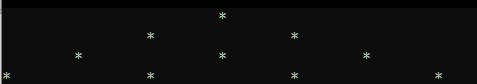
Eg31. Special pattern(hw=loops=pro12)

int main()

{

int i,j,k;

for(i=1;i<=4;i++)

 {

for(j=3;j>=i;j--)

{

printf("\t"); //or printf(" ");

}

for(k=1;k<=i;k++)

{

printf("\*\t\t"); // or printf("\* ");

}

printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**WHILE LOOP**

**Syntax**

**in main()**

**{**

**int i;i=1; // inilization;**

**while(i<=1) // while(condition)**

**{**

**printf("%d",&n); // statement;**

**i++; // updation;**

**}**

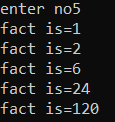
**}**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Eg32. Factorial no using only while(lecture=do while factorial)**

**int main()**

**{**

** int i,n,fact=1;**

**printf("enetr no:");**

**scanf("%d",&n);**

**i=1;**

**while(i<=n)**

**{**

**fact=fact\*i;**

**printf("fact is=%d\n",fact);**

**i++;**

**}**

**}**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DO WHILE LOOP**

Syntax

do

{

body of loop

}

while(condition);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg33. Factorial using do while(lecture=do while factorial2)

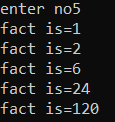
int main()

{

int i,fact=1,n;

printf("enter no");

scanf("%d",&n);

 i=1;

do

{

fact=fact\*i;

printf("fact is=%d\n",fact);

i++;

}

while(i<=n);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

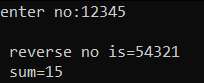
* Like for and while loop, do while is also used for executing a set of statements again &again by writing once. But unlike other loops it first executes the statements within the loop body and then checks condition. It always executes at least once.

**REVERSING NUMBERS USING WHILE**

Eg34. (lecture=loops=revsersing no using while loop)

int main()

{

 int n,rev=0,r,sum=0;

printf("enter no:");

scanf("%d",&n);

while(n!=0)

{

r=n%10;

rev=rev\*10+r;

n=n/10;

sum=sum+r;

}

printf("\n reverse no is=%d",rev);

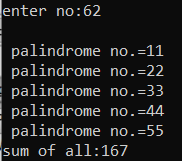
printf("\n sum=%d",sum);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PALINDROME NUMBERS(11,22,33……)**

Eg35.(lecture=loops=palindrome numbers)

int main()

{

int i,n,rev=0,sum=0,photo,r=0;

printf("enter no:");

scanf("%d",&n);

for(i=11;i<=n;i++)

{

photo=i;

rev=0;

while(photo!=0)

{

r=photo%10;

photo=photo/10;

rev=rev\*10+r;

sum=sum+photo;

}

if(i==rev)

{

printf("\n palindrome no.=%d",i);

}

}

printf("\nsum of all:%d",sum);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ARRAY**

Eg.36 basics using salary example(lecture=array=basic)

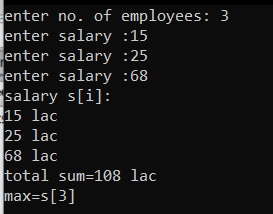
int main()

{

int sum=0,i,n,max;

printf("enter no. of employees: ");

scanf("%d",&n);

 int s[n];

for(i=0;i<n;i++)

{

printf("enter salary :",i);

scanf("%d",&s[i]);

}

printf("salary s[i]:\n");

for(i=0;i<n;i++)

{

printf("%d lac\n",s[i]);

sum=sum+s[i];

}

{

printf("total sum=%d lac\n",sum);

}

max=s[0];

for(i=0;i<n;i++)

{

if(max<s[i]) // min>s[i]

{

max=s[i];

}

}

printf("max=s[%d]\n",i);

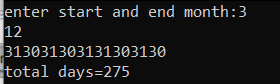
}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

Eg37. Adding no of days using starting and ending months(hw=pro=pro4)

int main()

{

int days[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};

 int sm,em,total=0,i;

printf("enter start and end month:");

scanf("%d %d",&sm,&em);

for(i=sm;i<em;i++)

{

printf("%d",days[i]);

total=total+days[i];

}

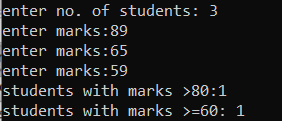
printf("\ntotal days=%d",total);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg38. No of products b/w other similar(hw=pro=array=pro2)

int main()

{

int i,n,count=0,c=0;

printf("enter no. of students: ");

scanf("%d",&n);

int s[n];

for(i=0;i<n;i++)

{

printf("enter marks:");

scanf("%d",&s[i]);

if(s[i]>80)

{

c++;

}

else

if(s[i]>60 && s[i]<70)

{

count++;

}

}

printf("students with marks >80:%d \n",c);

printf("students with marks >=60: %d",count);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Searching elements in array**

Eg39.(lecture=array=earching elements)

int main()

{

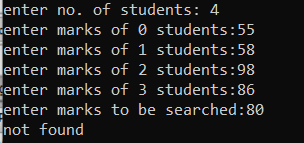
int n,i,wanted,count=0;

printf("enter no. of students: ");

scanf("%d",&n);

int m[n];

for(i=0;i<n;i++)

 {

printf("enter marks of %d students:",i);

scanf("%d",&m[i]);

}

printf("enter marks to be searched:");

scanf("%d",&wanted);

for(i=0;i<n;i++)

{

if(wanted==m[i])

{

printf("found at %d location:",i);

count++;

}

}

if(count==0)

printf("not found");

else printf("found %d times",count);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SORTING**

Eg40.sorting data(c=lecture=array=sorting element)

int main()

{

int n,r,c,temp;

printf("enter no.");

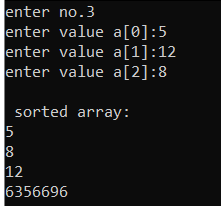
scanf("%d",&n);

int a[n];

for(r=0;r<n;r++)

{

printf("enter value a[%d]:",r);

 scanf("%d",&a[r]);

}

for(r=n-2;r>=0;r--)

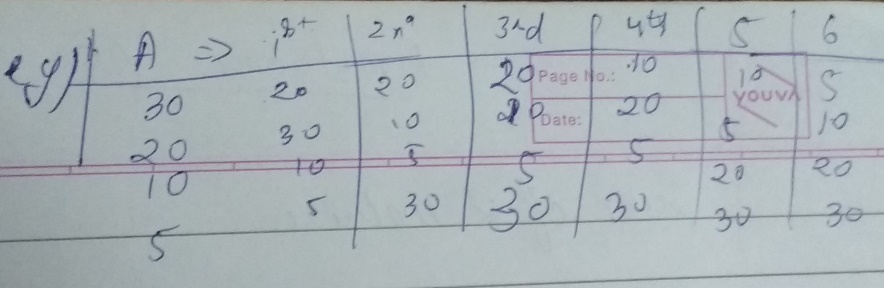
{

for(c=0;c<=r;c++)

{

if(a[c]>a[c+1])

{

 temp=a[c];

a[c]=a[c+1];

a[c+1]=temp;

}

}

}

printf("\n sorted array:\n");

for(r=0;r<=n;r++)

{

printf("%d\n",a[r]);

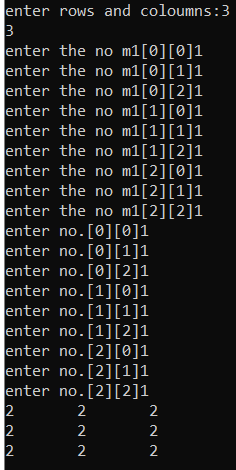
}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Sum of 2 matrix**

Eg41. sum of 2 matrix(c=lecture=array=sum of 2 matrix)

int main()

{

int m,n,r,c;

printf("enter rows and coloumns:");

scanf("%d%d",&m,&n);

int m1[m][n],m2[m][n],m3[m][n];

for(r=0;r<m;r++)

{

for(c=0;c<n;c++)

{

printf("enter the no m1[%d][%d]",r,c);

scanf("%d",&m1[r][c]);

}

}

for(r=0;r<m;r++)

{

for(c=0;c<n;c++)

{

printf("enter no.[%d][%d]",r,c);

scanf("%d",&m2[r][c]);

}

}

for(r=0;r<m;r++)

{

for(c=0;c<n;c++)

{

m3[r][c]=m1[r][c]+m2[r][c];

printf("%d\t",m3[r][c]);

}

printf("\n");

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Counting no of applicants in certain limit**

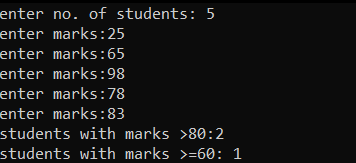
Eg42.(c=hw=program=array=pro2)

int main()

{

int i,n,count=0,c=0;

printf("enter no. of students: ");

 scanf("%d",&n);

int s[n];

for(i=0;i<n;i++)

{

printf("enter marks:");

scanf("%d",&s[i]);

if(s[i]>80)

{

c++;

}

else

if(s[i]>60 && s[i]<70)

{

count++;

}

}

printf("students with marks >80:%d \n",c);

printf("students with marks >=60: %d",count);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg43. Sum of 2 arrays(hw=a­­­­rray=pro3)

int main()

{

int i,n,sum=0;

printf("enter no.: ");

scanf("%d",&n);

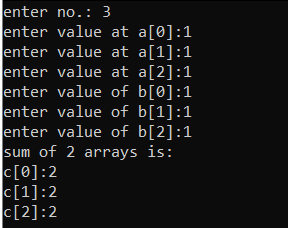
int a[n];

for(i=0;i<n;i++)

{

printf("enter value at a[%d]:",i);

scanf("%d",&a[i]);

 }

int b[n];

for(i=0;i<n;i++)

{

printf("enter value of b[%d]:",i);

scanf("%d",&b[i]);

}

printf("sum of 2 arrays is: \n");

int c[i];

for(i=0;i<n;i++)

{

c[i]=a[i]+b[i];

printf("c[%d]:%d\n",i,c[i]);

}

}

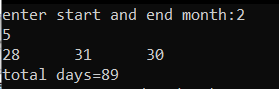
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg44. Sum of days b/w 2 months(c=hw=program=pro4)

int main()

{

int days[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};

 int sm,em,total=0,i;

printf("enter start and end month:");

scanf("%d %d",&sm,&em);

for(i=sm;i<em;i++)

{

printf("%d",days[i]);

total=total+days[i];

}

printf("\ntotal days=%d",total);

}  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg45. Armstrong numbers(c=lecture=array=Armstrong no.)

int main()

{

int r,n1,sum=0,original=0;

 printf("enter no.");

scanf("%d",&n1);

original=n1;

while(r!=0)

{

r=n1%10;

n1=n1/10;

sum=sum+r\*r\*r;

}

if(original==sum)

printf("armstrong");

else

printf(" non armstrong no.");

}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg45.multiplication of 2 matrix(c=lecture=array=multiplying of 2 matrix)

int main()

{

int ra,ca,rb,cb,r,c,k,sum=0;

printf("enter order of matrix a=");

scanf("%d%d",&ra,&ca);

printf("enter order of matrix b=");

scanf("%d%d",&rb,&cb);

int a[ra][ca],b[rb][cb],ab[10][10];

if(ca!=rb)

{

printf("invalid order");

}

else

{

for(r=0;r<ra;r++)

{

for(c=0;c<ca;c++)

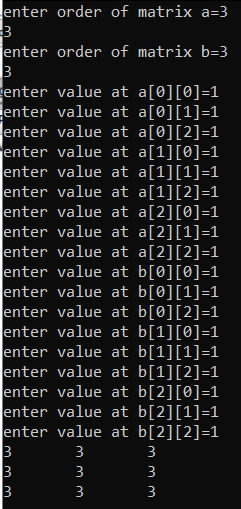
{

printf("enter value at a[%d][%d]=",r,c);

scanf("%d",&a[r][c]);

}

}



for(r=0;r<rb;r++)

{

for(c=0;c<cb;c++)

{

printf("enter value at b[%d][%d]=",r,c);

scanf("%d",&b[r][c]);

}

}

for(r=0;r<ra;r++)

{

for(c=0;c<cb;c++)

{

sum=0;

for(k=0;k<ca;k++)

{

sum=sum+a[r][k]\*b[k][c];

ab[r][c]=sum;

}

}

}

for(r=0;r<ra;r++)

{

for(c=0;c<cb;c++)

{

printf("%d\t",ab[r][c]);

}

printf("\n");

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

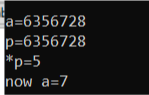
**POINTERS**

-A pointer is a type of a special variable which can hold the address of variable.

-using a pointer variables we can access and modify the value of a variable of which it can contain the address.

Eg46. basic(lecture=pointers=basic)

int main()

{

int a=5;

int \*p; //declaration of pointer variables

p=&a;

printf("\na=%d",&a);//100

printf("\np=%d",p);//100

printf("\n\*p=%d",\*p);//5

\*p=7; //modification

printf("\nnow a=%d",a);//7

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- we cant perform any arithmetic operation on address except subtraction.

Int a=5,b=6,\*p;

Printf(“%d”,&a+&b); ×

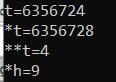
Printf(“%d”,&a\*&b); ×

Printf(“%d”,&a-&b); ☑

Printf(“%d”,&a/&b); ×

**POINTER TO POINTER**

Eg.47 (lecture=pointer=pointer to pointer)

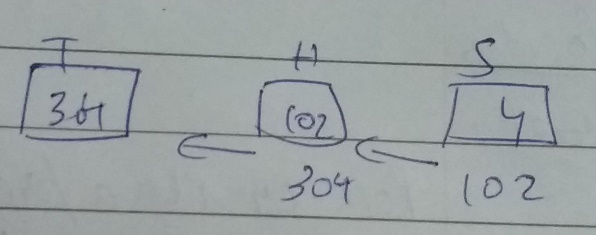
int main()

{

int s=4,\*h,\*\*t;

h=&s;

t=&h;

 printf("\nt=%d",t);//304

printf("\n\*t=%d",\*t);//102

printf("\n\*\*t=%d",s);//4

\*\*t=9;//modify

printf("\n\*h=%d",\*h);//9

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Pointer to array**

Point containing base address of array is known as pointer to array. The address of 0th location of array base address of array

Eg48.(lecture=pointer=pointer to array)

int main()

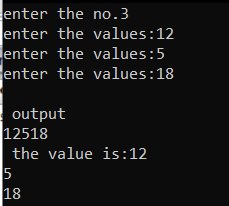
{

int n,i;

printf("enter the no.");

scanf("%d",&n);

int a[n],\*p;

 for(i=0;i<n;i++)

{

printf("enter the values:",i);

scanf("%d",&a[i]);

}

p=&a[0];

printf("\n output\n");

for(i=0;i<n;i++)

{

printf("%d",a[i]);//or printf("%d",\*(p+i));

}

\*(p+i)=\*((p+i)+10);//modification

printf("\n the value is:");

for(i=0;i<n;i++)

printf("%d\n",a[i]);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**FUNCTIONS**

syntax. basics(lecture=function=basic)

int main()

{

void rect(); //fun. declaration

rect; //fun calling

printf("back in main"); //fun. defination

}

void rect()

{

int l,b,a;

printf("enter l and b:");

scanf("%d %d",&l,&b);

a=l\*b;

printf("area is=%d",a);

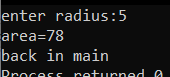
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg49. using formula in functions(lecture=function=using formula)

int main()

{

 void area();

area();

printf("back in main");

}

void area()

{

int r,area;

printf("enter radius:");

scanf("%d",&r);

area=3.14\*r\*r;

printf("area=%d\n",area);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**FUNCTION WITH ARGUMENT**

Usage= reusing data

Eg50. Basic(lecture=function=function with argument)

int main()

{

 void rect(int l, int b);

int ln;int b;

printf("enter the l&b:");

scanf("%d%d",&ln,&b);

rect(ln,b);//actual agrument

}

void rect (int l, int b)

{

int a=l\*b;

printf("area=%d",a);

}

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

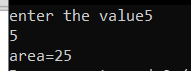
**Function with argument with return**

Eg51. (lecture=function=fun with argument with return)

int main()

{

Int ~~void~~ rect(int l, int b);//[fx prototype/signature],[int=> return type][int l,int b=>argument types]

 int l,b;

printf("enter the value");

scanf("%d%d",&l,&b);

int ar=rect(l,b);//[actual argument/ parameter]

printf("area=%d",ar);

}

int rect(int l, int b)//[fx with argument],[ int l, int b=> formal arguments(recievers)]

{

int a;

a=l\*b;

return(a);

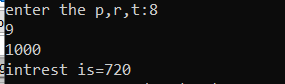
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg51.b Finding interest (lecture=function=function with argument intrest)

int main()

{

 void intrest(int p, int r, int t);

int p; int r; int t;

printf("enter the p,r,t:");

scanf("%d%d%d",&p,&r,&t);

intrest(p,r,t);//actual agrument

}

void intrest (int p, int r, int t)

{

int i=(p\*r\*t)/100;

printf("intrest is=%d",i);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TYPES OF PARAMETERS**

Call by value

* In call by value a copy of actual argument is passed to formal argument variable. if any changes remains using formal argument variables. It will not change original value of actual argument.

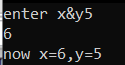
Eg52. (lecture=function=call by value swapping)

int main()

{

void swap(int a, int b);

int x,y;

 printf("enter x&y");

scanf("%d%d",&x,&y);

swap(x,y);

}

void swap(int a,int b)

{

int temp=a;

a=b;

b=temp;

printf("now x=%d,y=%d",a,b);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call by reference

* Call by reference of c code in c++ is called by address. Here the address of actual argument variable are pass to pointer variables taken as formal. If any changes made in the value using these pointers variables it will change the original values of actual argument variables.

Eg53. (lecture=function=call by referenceswapping)

int main()

{

 void swap(int \*a, int \*b);

int x,y;

printf("enter x&y");

scanf("%d%d",&x,&y);

swap(&x,&y);

printf("now x=%d y=%d",x,y);

}

void swap(int \*a, int\*b)

{

int temp=\*a;

\*a=\*b;

\*b=temp;

}

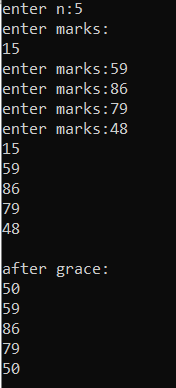
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PASSING ARRAY AS ARGUMENT**

Eg54. (lecture=function= passing arra as aargument(pointer to array))(grace marks)

int main()

{

 void grace(int \*p,int n);

int n,i;

printf("enter n:");

scanf("%d",&n);

int marks[n];

for(i=0;i<n;i++)

{

printf("enter marks:",i);

scanf("%d",&marks[i]);

}

grace(&marks[0],n);

printf("\nafter grace:\n");

for(i=0;i<n;i++)

{

printf("%d\n",marks[i]);

}

}

void grace(int \*p,int n)

{

int i;

for(i=0;i<n;i++)

{

printf("%d\n",\*(p+i));

if(\*(p+i)<50)

{

\*(p+i)=50;

}

}

}

When we pass array as argument basically we pass base address of array.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**STORAGE CLASS**

**Local/auto variables-**variables declaration within function body are known as local variables. It takes garbage vlues by default. It is not accessible in other functions.

Syntax

int main()

{

auto int x;//local variables

void show ()

printf("%d",x);

x=10;

show();

}

void show()

{

printf("%d",x);//error (not accesible)

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Glogal variables-global variables declare outside function body are know as global variables. These are accessible within all fun in the program. It takes 0 value by default.

Syntax

int x

int main()

{

void show ()

printf("%d",x);

show();

}

void show()

{

printf("%d",x);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**STATIC VARIABLES**

Static variables retain their value from previous certain their value call.

Syntax

int main()

{

void show()

show();

show();

}

void show()

{

static int x=5;

printf("x=%d",x)

x=x+10;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**REGISTER VARIABLES**

Using keyword register occupy storage space within cpu register instead o ram. It will increase the performance of cpu

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**RECURSSION**

When a function calls itself then recursion takes place.

Syntax

Int main()

{

Int i=0;

Printf(“enter N:”);

I++;

If(i<=3)

Main();

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Factorial of n using recursion**

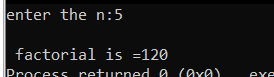
Eg55. (lecture=function=factorial using recursion)

int main()

{

int factorial(int n);

int n,fact;

 printf("enter the n:");

scanf("%d",&n);

fact=factorial(n);

printf("\n factorial is =%d",fact);

}

int factorial(int n)

{

int fact;

if (n==1)

return(1);

else

fact=factorial(n-1)\*n;

return(fact);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**STRINGS**

**Syntax**

int main()

{

char name[10]="bce";//Declaration Cum Inilization of string variable

printf("name==%s\n",name);

puts(name);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

OR

int main()

{

char name[30];

int i,cout=0;

printf("enetr the string");

gets(name);

printf("\n your name=%s",name);

}

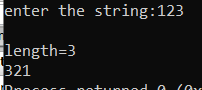
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Reverse of a string**

Eg56. (lecture=strings=reverse of string)

int main()

{

 char s[100];

int i,count=0;

printf("enter the string:");

gets(s);

for(i=0;s[i]!='\0';i++)

{

count++;

}

printf("\nlength=%d\n",count);

for(i=count-1;i>=0;i--)

{

printf("%c",s[i]);

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

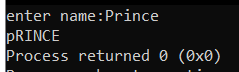
**CHANGING INTO UPPER OR LOWER CASES**

Eg57.(lecture=strings=changing into upper or lower case)

int main()

{

char a[20];

 printf("enter name:");

gets(a);

int i;

for(i=0;a[i]!='\0';i++)

{

if(a[i]>=65 && a[i]<=90)

a[i]=a[i]+32;

else

{

if(a[i]>=97 && a[i]<=122)

a[i]=a[i]-32;

}

}

printf("%s",a);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

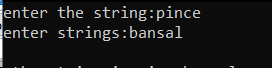
**JOINING OF ARRAY**

Eg58.(lecture=strings=joining 2strings)

int main()

{

char s1[50],s2[50],i,j;

 printf("enter the string:");

scanf("%s",s1);

printf("enter strings:");

scanf("%s",s2);

for(i=0;s1[i]!='\0';i++);

for(j=0;s2[j]!='\0';j++,i++)

{

s1[i]=s2[j];

}

s1[i]='\0';

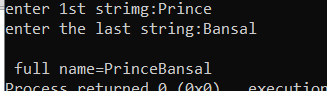
printf("\n the string is=%s",s1);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg59.(lecture=strings=joining 2stringstype2)

int main()

{

char s1[50],s2[50];

printf("enter 1st strimg:");

gets(s1);

printf("enter the last string:");

gets(s2);

strcat(s1,s2);

printf("\n full name=%s",s1);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**LIBRARY FUNCTIONS**

Eg60.(lecture=strings= LibraryFunctions)

int main()

{

char s2[10];

char s1[10]="bce";

int l=strln(s2);

printf("length=%d\n",l);

strln(s1);

printf("\nlower=%s",s1);

strupr(s1);

printf("\nupper case=%s",s1); Just For Knowledge

strcpy(s2,s1);

printf("\ncopied=%s",s2);

strrev(s2);

printf("rev=%s",s2);

int (rev)=strcmp(s1,s2);

if(rev==0)

printf("palindrome");

else

printf("not");

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**VOWELS IN A NAME**

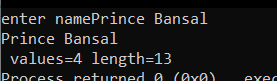
Eg61.(lecture=string=vowels)

int main()

{

int i,v=0;

char name[20];

 printf("enter name");

gets(name);

for(i=0;name[i]!='\0';i++)

{

printf("%c",name[i]);

switch (name[i])

{

case'a':case'e':case'i':case'o':case'u':

v++;

}

}

printf("\n values=%d length=%d",v,i);

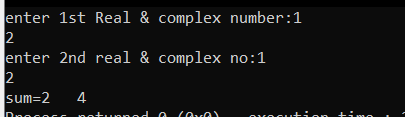
}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**STRUCTURES**

Eg62.(lecture=structure=basics)

struct complex// a user defined data type

{

int r,i;

};

int main()

{

struct complex c1,c2,c3;

printf("enter 1st Real & complex number:");

scanf("%d %d",&c1.r,&c1.i);

printf("enter 2nd real & complex no:");

scanf("%d %d",&c2.r,&c2.i);

c3.r=c1.r+c2.r;

c3.i=c1.i+c2.i;

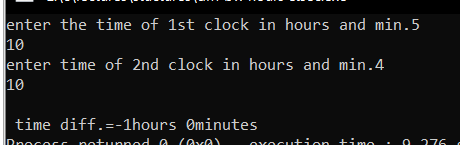
printf("sum=%d %d",c3.r,c3.i);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg63.difference b/w the hours and sec(lecture=structure= diff bw hours &sec)

struct clock

{

int h,m;

};

int main()

{

struct clock c1,c2,c3;

printf("enter the time of 1st clock in hours and min.");

scanf("%d %d",&c1.h,&c1.m);

printf("enter time of 2nd clock in hours and min.");

scanf("%d %d",&c2.h,&c2.m);

int df=((c2.h\*60+c2.m)-(c1.h\*60+c1.m));

c3.h=df/60;

c3.m=df%60;

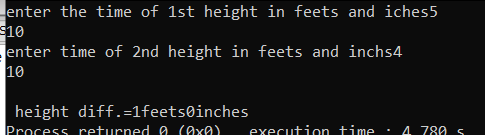
printf("\n time diff.=%dhours %dminutes",c3.h,c3.m);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eg64.difference b/w the feets and inches(lecture=structure= dif in feet and inches)

struct height

{

int f,i;

};

int main()

{

struct height c1,c2,c3;

printf("enter the time of 1st height in feets and iches");

scanf("%d %d",&c1.f,&c1.i);

printf("enter time of 2nd height in feets and inchs");

scanf("%d %d",&c2.f,&c2.i);

int df=abs((c2.f\*12+c2.i)-(c1.f\*12+c1.i));

c3.f=df/12;

c3.i=df%12;

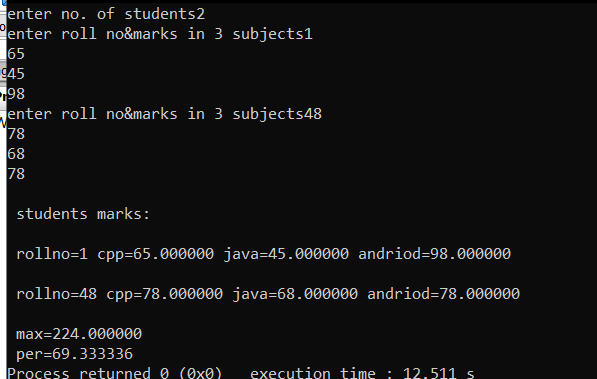
printf("\n height diff.=%dfeets%dinches",c3.f,c3.i);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ARRAY OF STRUCTURES**

Eg65. Max& percentage from marks(lecture=structures= array of structures)

struct student

{

int rollno;

float cpp,java,andriod,total,per;

};

int main()

{

int n,i;

printf("enter no. of students");

scanf("%d",&n);

struct student s[n];

for(i=0;i<n;i++)

{

printf("enter roll no&marks in 3 subjects");

scanf("%d%f%f%f",&s[i].rollno,&s[i].cpp,&s[i].java,&s[i].andriod);

s[i].total=s[i].cpp+s[i].java+s[i].andriod;

s[i].per=s[i].total\*100/300;

}

printf("\n students marks:\n");

float max=s[0].total;

float per=s[0].per;

for(i=0;i<n;i++)

{

printf("\n rollno=%d cpp=%f java=%f andriod=%f \n",s[i].rollno,s[i].cpp,s[i].java,s[i].andriod);

if(max<s[i].total)

{

if(per>s[i].per)

{

per=s[i].per;

}

max=s[i].total;

}

}

printf("\n max=%f",max);

printf("\n per=%f",per);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**POINTER TO STRUCTURE**

A pointer which can store the address of variable of structure of type is known as pointer to structure. Using a pointer to structure we can access and modify the whole number variables.

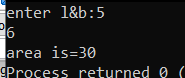
Eg66.(lecture=structure= pointer to structure)

struct rect

{

int l,b,a;

};

int main()

{

struct rect r;

struct rect\*p;// dec. of pointer to structure

p=&r;

printf("enter l&b:");

scanf("%d %d",&r.l,&r.b);

(\*p).a=(\*p).l\*(\*p).b;//or p->=p->l\*p->b;

printf("area is=%d",p->a);

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***THE END***