**Program-1:Program to calculate area and circumference of the circle**

#include<stdio.h>

int main()

{

const float PI=3.14;

int r;//radius of the circle

float a, c;

printf("enter radius of circle :" );

scanf("%d", &r);

a = PI \* r \* r;//area of the circle formula

printf("Area of circle :%f\n",a);

c = 2 \* PI \* r;//circumference of the circle formula

printf("Circumference :%f",c);

return 0;

}

**Program-2:Program to explain all datatypes including signed and unsigned**

#include<stdio.h>

int main()

{

int i;//declaration and definition of integer

i=20;//initialization

float f;//declaration and definition of float

f=3.14;//initialization

char a;//declaration and definition of float

a='z';//initialization

double d;//declaration and definition of double

d=4.7;//initialization

long double ld;//declaration and definition of long double

ld=3.14e+2;//initialization

short int si;//declaration and definition of short int

si=-564;//initialization

long int li;//declaration and definition of long int

li=4345567;//initialization

unsigned int ui;//declaration and definition of unsigned int

ui=65;//initialization

signed int sgi;//declaration and definition of signed int

sgi=6776;//initialization

printf("size of integer :%ld\n" , sizeof(i)); //range:-32,768 to 32,767

printf("size of float :%ld\n" , sizeof(f)); //range: 3.4E +/-38

printf("size of character :%ld\n" , sizeof(a)); //range:-128 to 127

printf("size of double :%ld\n" , sizeof(d)); //range:1.7E +/- 308

printf("size of long double :%ld\n" , sizeof(ld)); //range:1.7E +/- 308

printf("size of short integer :%ld\n" , sizeof(si)); //range:-32,768 to 32,767

printf("size of long integer :%ld\n" , sizeof(li)); //range:-2,147,483,648 to -2,147,483,647

printf("size of unsigned integer :%ld\n" , sizeof(ui)); //range:0 to 65,535

printf("size of signed integer :%ld\n" , sizeof(sgi)); //range:-32,768 to 32,767

return 0;

}

**Program-3:Program to explain all operators**

#include<stdio.h>

int main()

{

int a, b;

float c,d;

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

scanf("%f %f\n",&c,&d);

//Arithmetic operators (+, -, \*, /, %)

printf("sum=%d\n", a+b); //4

printf("diff=%d\n", a-b); //0

printf("mul=%d\n", a\*b); //4

printf("div=%d\n", a/b); //1

printf("mod=%d\n", a%b); //0

//Relational operators(<, >, <=, >=, ==, !=)

printf("%d\n" ,c>d); //0

printf("%d\n" ,c<d); //0

printf("%d\n" ,c>=d); //1

printf("%d\n" ,c<=d); //1

printf("%d\n" ,c==d); //1

printf("%d\n" ,c!=d); //0

//Bitwise operators(&, |, ^, <<, >>)

printf("%d\n", a&b); //2

printf("%d\n", a|b); //2

printf("%d\n", a^b); //0

printf("%d\n", a<<b); //8

printf("%d\n", a>>b); //0

//logical operators(&&, ||, !)

printf("%d\n", a&&b); //1

printf("%d\n", a||b); //1

printf("%d\n", !a); //0

//assignment operators(+=, -=, \*=, /=, %=)

printf("%d\n", a+=b); //4

printf("%d\n", a-=b); //2

printf("%d\n", a\*=b); //4

printf("%d\n", a/=b); //2

printf("%d\n", a%=b); //0

//conditional operator to find largest number(?:)

printf("%d",((a>b)?a:b));

return 0;

}

**Program-4:Program to explain auto storage classes**

#include<stdio.h>

int fun()

{

auto int c=0;

c++;

return c;

}

int main()

{

printf("%d" ,fun());

printf("%d",fun());

return 0;

}

**Program-5:Program to explain extern storage class**

First file: main.c

#include<stdio.h>

extern i;

main()

{

printf(“value of the external integer is = %d\n” , i);

return 0;

}

Second file : orginial.c

#include<stdio.h>

i= 48;

**Program-6:Program to explain static local storage class**

#include<stdio.h>

int fun()

{

static int c=0;

c++;

return c;

}

int main()

{

printf(“ %d” , fun());

printf(“%d”, fun());

return 0;

}

**Program-7:Program to explain static global storage class**

#include<stdio.h>

static int c=0;

int fun()

{

c++;

return c;

}

int main()

{

printf(“ %d” , fun());

printf(“%d”, fun());

return 0;

}

**Program-8:Program to explain register storage class**

#include<stdio.h>

int main()

{

register int i;

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

printf(“%d” , i);

return 0;

}

**Program-9:Program to create lisnked list of 3 nodes**

#include<stdio.h>

#include<stdlib.h>

struct ll //linked list

{

int data;

struct ll \*next;

};

typedef struct ll node;

node \*first = NULL, \*last = NULL;

void create\_node(int ele)

{

node \*new = (node\*)malloc(sizeof(node));

if(first == NULL)

{

new -> data = ele;

new-> next = NULL;

first = last = new;

}

else

{

new -> data = ele;

new -> next = NULL;

last -> next = new;

last = new;

}

}

void display()

{

node \*temp = first;

if(temp==NULL)

{

printf("List is empty\n");

}

else

{

while(temp!=NULL)

{

printf("%d->",temp->data);

temp=temp->next;

}

}

}

int main()

{

create\_node(100);

create\_node(200);

create\_node(300);

display();

return 0;

}

**Program-10:Program to create a circular LL with 5 nodes having char data vowels –a,e,i,o,u**

#include<stdio.h>

#include<stdlib.h>

struct cll //circular linked list

{

char ch;

struct cll \*next;

};

typedef struct cll node;

node \*first = NULL, \*last = NULL;

void create\_node(char vowels)

{

node \*new = (node\*)malloc(sizeof(node));

if(first==NULL)

{

new -> ch=vowels;

new-> next = first;

first = last = new;

}

else

{

new -> ch = vowels;

new -> next = first;

last -> next = new;

last = new;

}

}

void display()

{

node \*temp = first;

do

{

printf("%c ", temp -> ch);

temp = temp -> next;

}while(temp!= first);

}

int main()

{

create\_node('a');

create\_node('e');

create\_node('i');

create\_node('o');

create\_node('u');

display();

return 0;

}