

### **Ex. No. 1 AREA OF TRIANGLE**

**AIM:** To write a program for evaluating the area of triangle using the formula  $\sqrt{s(s-a)(s-b)(s-c)}$ .

#### **ALGORITHM:**

Step1: Start the program.

Step2: Get the inputs a, b, c and s.

Step3: Calculate  $s = (a+b+c) / 2$ .

Step4: Calculate  $\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$ .

Step 5: Print the result 'area'.

Step 6: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<math.h>
void main()
{
int a,b,c;
float s,area;
clrscr();
printf("Enter the values of a,b,c: ");
scanf("%d%d%d",&a,&b,&c);
s=(a+b+c)/2;
area=sqrt(s*(s-a)*(s-b)*(s-c));
printf("The area of a triangle is =%f",area);
getch();
}
```

#### **OUTPUT:**

Enter the values of a,b,c: 10 20 30

The area of a triangle is = 0.000000

**RESULT:**

Thus the C program to find the area of triangle using the formula  $\sqrt{s(s-a)(s-b)(s-c)}$  has

been successfully executed and verified.

**Ex. No. 2 SWAP TWO NUMBERS**

**AIM:**To write a program for swapping of two numbers.

**ALGORITHM:**

Step1: Start the program.

Step2: Get the inputs a and b.

Step3: Find  $a=a+b$ .

Step4: Find  $b=a-b$ .

Step 5: Find  $a=a-b$ .

Step6: Print the result 'a' and 'b'.

Step7: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf("Enter the values of a and b: ");
scanf("%d%d",&a,&b);
a=a+b;
b=a-b;
a=a-b;
printf("The values of a and b are: %d %d", a, b);
getch();
}
```

**OUTPUT:**

Enter the values of a and b: 10 20

The values of a and b are: 20 10

**RESULT:**

Thus the C program to swap two numbers has been successfully executed and verified.

**Ex. No. 3 GREATEST OF THREE NUMBERS AND PRINT ASCENDING ORDER**

**AIM:**To write a program for finding the greatest of three numbers and printing the numbers in ascending order.

**ALGORITHM:**

Step1: Start the program.

Step2: Get the inputs a, b and c.

Step3: Check if((a>b) &&(a>c))

Step4: Again check if(b>c)

Step5: Then print the greatest number and display a, b, c.

Step6: Else print the greatest number and display a, c, b.

Step7: Check if((b<c) &&(b<a))

Step8: Again check if(c<a)

Step9: Then print the greatest number and display b, c, a.

Step10: Else print the greatest number and display b, a, c.

Step11: Check if((c<a) && (c<b))

Step12: Again check if(a<b)

Step13: Then print the greatest number and display c, a, b.

Step14: Else print the greatest number and display c, b, a.

Step15: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf("Enter the values of a, b and c: ");
scanf("%d%d%d", &a, &b, &c);
```

```
if(a<b && a<c)
{
if(b<c)
{
printf("The greatest number is: %d", a);
printf("The ascending order: %d%d%d", a, b, c);
}
else
if(b>c)
{
printf("The greatest number is: %d", a);
printf("The ascending order: %d%d%d", a, c, b);
}
} else if(b<c
&& b<a)
{
if(c<a)
{
printf("The greatest number is: %d", b);
printf("The ascending order: %d%d%d", b, c, a);
}
}
else
{
printf("The greatest number is: %d", b);
printf("The ascending order: %d%d%d", b, a, c);
}
}
else
```

```
if(b<a)
{
printf("The greatest number is: %d", c);
printf("The ascending order: %d%d%d", c, b, a);
} else
{
printf("The greatest number is: %d", c);
printf("The ascending order: %d%d%d", c, a, b);
}
}
```

**OUTPUT:** Enter the values of a, b and c: 6 4 5

The greatest number is: 6

The ascending order: 4 5 6

**RESULT:**

Thus the C program to find greatest of three and to print the numbers in ascending order

has been successfully executed and verified.

#### **Ex. No. 4 ARITHMETIC EXPRESSION USING SWITCH STATEMENT**

**AIM:**To write a program for performing the arithmetic expression using switch statement.

#### **ALGORITHM:**

Step1: Start the program.

Step2: Display 1. Addition 2. Subtraction 3. Multiplication and 4. Division

Step3: Get the input a and b.

Step4: Get the choice.

Step5: Switch(result)

Step6: case '+': print the sum of a & b.

Step7: case '-': print the difference of a & b.

Step8: case '\*': print the multiplication of a & b.

Step9: case '/': print the division of a & b.

Step10: default: invalid option.

Step11: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
int op;
clrscr();
printf("Enter the values of a & b: ");
scanf("%d%d", &a, &b);

printf(" 1.Addition\n 2.Subtraction\n 3.Multiplication\n 4.Division\n");

printf("Enter your choice: ");
```



```
scanf("%d", &op);
switch(op)
{
case 1 :printf("Sum of %d and %d=%d", a, b, a+b);
break;
case 2 :printf("Subtraction of %d and %d=%d", a, b, a-b);
break;
case 3 :printf("Multiplication of %d and %d=%d", a, b, a*b);
break;
case 4 :printf("Division of %d and %d=%d", a, b, a/b);
break;
default : printf(" Enter Your Correct Choice.");
break;
}
getch();
}
```

**OUTPUT:**

Enter the values of a & b: 10 20

1. Addition
2. Subtraction
3. Multiplication
4. Division

Enter your choice: 1

Sum of 10 and 20 = 30

**RESULT:**

Thus the C program for arithmetic expression using switch statement has been successfully executed and verified.

**Ex. No. 5 FACTORIAL OF A NUMBER USING DO WHILE STATEMENT**

**AIM: To write a program for finding the factorial of a given number using do while statement.**

**ALGORITHM:**

Step1: Start the program.

Step2: Assign  $f=i=1$ .

Step3: Get the input n.

Step4: do .. the following.

Step5: Find  $f=f*i$

Step6: Increment  $i=i+1$

Step7: Repeat from step5 to step6 till while( $i \leq n$ ).

Step8: Then print f.

Step9: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i,f;
f=i=1;
clrscr();
printf("Enter a number: ");
scanf("%d",&n);
do
{
fact*=i;
i++;
}while(i<=n);
```

```
printf("Factorial of %d=%d\n", no, fact;
}
```

**OUTPUT:**

Enter a number: 5

Factorial of 5 = 120

**RESULT:**

Thus the C program for finding the factorial of a given number using do while statement has been successfully executed and verified.

**Ex. No. 6 GENERATE PRIME NUMBERS UPTO N NUMBERS**

**AIM: To write a program for printing all prime numbers upto N numbers.**

**ALGORITHM:**

Step1: Start the program.

Step2: Get the n value.

Step3: for(i=1;i<=n;i++)

Step4: Repeat a, b, c, d & e

a) Assign fact=0

b) for(j=1;j<=n;j++) repeat c & d

c) if i percentage j equal to zero

d) fact equal to fact added with one

e) if fact equal to 2 print i as prime number

Step5: Display the prime number till nth number.

Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i,fact,j;
printf("Enter the range: ");
scanf("%d",&n);
printf("Prime numbers are: \n");
for(i=1;i<=n;i++)
{
fact=0;
for(j=1;j<=n;j++)
{
```

```
if(i%j==0)
fact++;
if(f==2)
printf("%d ",i);
}
getch();
}
```

**OUTPUT:**

Enter the range: 10

Prime numbers are: 3 5 7

**RESULT:**

Thus the C program for printing all prime numbers upto N numbers has been successfully executed and verified.

**Ex. No. 7 SUM OF N NATURAL NUMBERS**

**AIM: To write a program for printing the sum of N natural numbers.**

**ALGORITHM:**

Step1: Start the program.

Step2: Get the n value.

Step3: Initialize i=0 and sum=0.

Step4: Perform from step 5 to step 6 until i<=n

Step5: i++

Step6: sum+=i

Step7: Print the sum.

Step9: Stop the program.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i=0,sum=0;
clrscr( );
printf("Enter the Limit : ");
scanf("%d",&n);
while(i<=n)
{
i++;
sum+=i;
}
printf("Sum of %d natural numbers = %d",n,sum);
getch();
}
```

**OUTPUT:**

Enter the Limit : 10

Sum of 10 natural numbers = 55

**RESULT:**

Thus the C program for printing the sum of N natural numbers has been successfully executed and verified.

**Ex. No. 8 TOTAL NUMBER OF EVEN INTEGERS AND ODD INTEGERS OF 'N' NUMBERS**

**AIM: To write a program for finding the total number of even integers and odd integers of 'N' numbers.**

**ALGORITHM:**

Step1: Start the program.

Step2: Declare int i, n, odd=0 and even=0;

Step3: Get the n value

Step4: for( i=0;i<=n;i++) do the following step.

a) Check if(i%2==0)

b) even=even+1;

c) Else odd=odd+1;

Step5: Print the odd and even value.

Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i,odd=0,even=0;
clrscr();
printf("Enter the n value: ");
scanf("%d",&n);
for(i=1;i<n;i++)
{
if(i%2==0)
even=even+1;
else
```



```
odd=odd+1;
}
printf("The total number of odd integers =%d",odd);
printf("The total number of even integers =%d",even);
getch();
}
```

**OUTPUT:**

Enter the n value: 10

The total number of odd integers =5

The total number of even integers = 5

**RESULT:**

Thus the above C program for finding the total number of even integers and odd integers of 'N' numbers has been successfully executed and verified.

**Ex. No. 9 SUM OF EVEN INTEGERS AND ODD INTEGERS OF 'N' NUMBERS**

**AIM: To write a program for finding the sum of even integers and odd integers of 'N' numbers.**

**ALGORITHM:**

Step1: Start the program.

Step2: Declare int i, n, odd=0 and even=0;

Step3: Get the n value

Step4: for( i=0;i<=n;i++) do the following step.

a) Check if(i%2==0)

b) even=even+i;

c) Else odd=odd+i;

Step5: Print the odd and even value.

Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,n,sum,even=0,odd=0;
clrscr();
printf("Enter any number: ");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
if(i%2==0)
even=even+i;
else
odd=odd+i;
```

```
}  
printf("Sum of even integer is: %d",even);  
printf("Sum of odd integer is: %d",odd);  
getch();  
}
```

**OUTPUT:**

Enter any value: 5

Sum of even integer is: 6

Sum of odd integer is: 9

**RESULT:**

Thus the C program for finding the sum of even integers and odd integers of 'N' numbers has been successfully executed and verified.

### **Ex. No. 10 PRODUCT OF TWO MATRICES OF ANY ORDER**

**AIM: To write a program for finding the product of two matrices of any order.**

#### **ALGORITHM:**

Step1: Start the program.

Step2: Declare int Matrix A[9][9] , MatrixB[9][9] , Matrixsproduct [9][9].

Step3: Declare int n , i , j , k, Row1 , Row2 , Column1 , Column2.

Step4: Enter the order of Matrix A Row1, Column1.

Step4: Enter the order of Matrix B Row2, Column2.

Step5: Check if(Column1 == Row2)

Step6: Enter the elements of Matrix A and B using for loops.

Step7: Find Matrixproduct[i][j] = Matrixproduct[i][j] +(Matrix A[i][k]  
\*

Matrix B[k][j] using for loops.

Step7: Print the resultant matrix Matrixproduct[i][j] using for loop.

Step8: Else print invalid order so multiplication not possible.

Step9: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int Matrix A[9][9] , MatrixB[9][9] , Matrixsproduct [9][9] ;
int n , i , j , k; /* 'i' used for rows and 'j' used for columns */
int Row1 , Row2 , Column1 , Column2;
clrscr();
printf(" Enter the order of Matrix A\n");
scanf("%d * %d " , &Row1 , &Column1);
```

```
printf(" Enter the order of Matrix B\n");
scanf("%d * %d " , &Row2 , &Column2);
if(Column1 == Row2)
{
printf(" Enter the elements of Matrix A\n");
for(i=0 ; i<Row1 ; i++)
{
for(j=0 ; j<Column1 ; j++)
{
scanf("%d" , &Matrix A[i][j] );
}
}
printf(" Enter the elements of Matrix B\n");
for(i=0 ; i<Row2 ; i++)
{
for(j=0 ; j<Column2 ; j++)
{
scanf("%d" , &Matrix B[i][j] );
}
}
for(i=0 ; i<Row1 ; i++)
{
for(j=0 ; j<Column2 ; j++)
{
Matrixproduct[i][j] = 0 ;
for(k=0 ; k<Row2 ; k++)
{
```

```
Matrixproduct[i][j] = Matrixproduct[i][j] +(Matrix A[i][k] * Matrix
B[k][j] );
}
}
}
printf(" Product Matrix\n");
for(i=0 ; i< Row1 ; i++)
{
for(j=0 ;j< Column2;j++)
{
printf("%d" , Matrixproduct[i][j] );
}
printf("\n");
}
} else
printf(" Invalid order so Multiplication not possible\n");
}
```

**OUTPUT:**

Enter the order of Matrix A

2 \* 2

Enter the order of MatrixB

2 \* 2

Enter the elements of Matrix A

1234

Enter the elements of Matrix B

5678

Product Matrix

19 22

43 50

**RESULT:**

Thus the C program for finding the product of two matrices of any order has been successfully executed and verified.

**Ex. No. 11 READ 'N' NUMBER OF STUDENTS WITH 5 SUBJECT MARKS**

**AIM: To write a program for reading 'N' number of students with 5 subject marks.**

**ALGORITHM:**

Step1: Start the program.

Step2: Initialize a character array n and integer array r and s.

Step3: Initialize integer i, j and n.

Step3: Read the value of n.

Step4: for(i=0;i<n;i++)

a) Enter rollno,name,,,,,

b) Read these and enter 5 subject marks using for loop and array.

Step5: Display n[i],r[i],s[i][j]

Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
char n[20][10];
int i,j,r[20],s[20][6];
printf("Enter n value: ");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter name,rollno,....");
scanf("%s%d",&n[i],&r[i]);
printf("Enter 5 subject marks:");
s[i][5]=0;
```



```
for(j=0;j<5;j++)
{
scanf("%d",s[i][j]);
s[i][5]=s[i][5]+s[i][j];
}
}

printf("The data entered is: \n");
for(i=0;i<n;i++)
{
printf("%s\t%d\t",n[i],r[i]);
for(j=0;j<5;j++)
printf("%d\t",s[i][j]);
}
getch();
}
```

**OUTPUT:**

```
Enter n value: 1
Enter name,rollno,...Eswar 20
Enter 5 subject marks:
10 50 34 06 42
The data entered is:
Eswar 20 10 50 34 06 42
```

**RESULT:**

Thus the C program for reading 'N' number of students with 5 subject marks has been successfully executed and verified.

### **Ex. No. 12 GREATEST OF 'N' NUMBERS USING FUNCTION**

**AIM: To write a program for finding greatest of 'n' numbers using function.**

#### **ALGORITHM:**

Step1: Start the program.

Step2: Initialize integer a, b and c.

Step3: Read the value of a,b and c.

Step4: Call the function large().

a) Check if((a>b) && (a>c)) then print a is greater.

b) Check elseif (b>c) then print b is greater.

c) Check else print c is greater.

Step5: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
printf(" Enter the value of a,b and c: ");
scanf("%d, %d, %d", &a, &b, &c);
large(a,b,c);
getch();
}
large(int a, int b, int c)
{
if((a>b) && (a>c))
print("%d is greater than %d, %d", a, b, c);
elseif (b>c)
```

```
print("%d is greater than %d, %d", b, a, c);  
else  
print("%d is greater than %d, %d", c, a, b);  
}
```

OUTPUT:

Enter the value of a,b and c: 10 30 20

30 is greater than 10, 20

**RESULT:**

Thus the C program for finding greatest of 'n' numbers using function has been successfully executed and verified.

### **Ex. No. 13 FIBONACCI SERIES USING RECURSION**

**AIM: To write a program for finding Fibonacci series using recursion.**

#### **ALGORITHM:**

Step1: Start the program.

Step2: Initialize a function as `int Fibonacci(int)`.

Step3: Initialize integer `i=0`, `c` and `n` in main function.

Step3: Read the value of `n`.

Step4: Within for loop call the `Fibonacci(int)` recursively.

Step5: In `Fibonacci(int)` function calculate ( `Fibonacci(n-1)` + `Fibonacci(n-2)` )

recursively and return the value.

Step6: Print the result.

Step7: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>

int Fibonacci(int);

int main()
{
    int n, i = 0, c;
    printf("Enter the n value: ");
    scanf("%d",&n);
    printf("Fibonacci series\n");
    for ( c = 1 ; c <= n ; c++ )
    {
        printf("%d\n", Fibonacci(i));
        i++;
    } return 0;
}
```

```
int Fibonacci(int n)
{
if ( n == 0 )
return 0;
else if ( n == 1 )
return 1;
else
return ( Fibonacci(n-1) + Fibonacci(n-2) );
}
```

**OUTPUT:**

Enter the n value: 9

Fibonacci series: 0 1 1 2 3 5 8 13 21

**RESULT:**

Thus the C program for finding Fibonacci series using recursion has been successfully executed and verified.

### **Ex. No. 14 LOWER CASE TO UPPERCASE CHARACTERS**

**AIM: To write a program for converting all lower case to uppercase characters.**

#### **ALGORITHM:**

Step1: Start the program.

Step2: Take a string a function of return value data type is void str upper.

Step3: Read a string.

Step4: While (s[i] != '\0') the do the following

a) if((s[i] >= 'a') && (s[i] <= 'z'))

b) s[i] = s[i] - 32;

c) i++;

Step5: Display changed string.

Step6: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
char str;
printf("Enter a string: ");
scanf("%s",str);
to_str_upper(char[]);
printf("Changed to: %s",str);
}
void to_str_upper(char[])
{
int i=0;
while(s[i]!='\0')
```

```
{  
if((s[i]>='a') && (s[i]>='z'))  
s[i]=s[i]-32;  
i++;  
}  
}
```

**OUTPUT:**

Enter a string : g nec changed to: GNEC

**RESULT:**

Thus the C program for converting all lower case to uppercase characters has been successfully executed and verified.

**Ex. No. 15 SORT 5 CITY NAMES IN ALPHABETICAL ORDER**

**AIM: To write a program for sorting 5 city names in alphabetical order.**

**ALGORITHM:**

Step1: Start the program.

Step2: Using for loop and array get the city name.

Step3: Using loop for(i=65;i<122;i++) and for(j=0;j<5;j++)

a) Check if(city[j][0]==i)

b) Display the sorted list of cities.

Step4: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void main()
{
ch city[5][20];
int i,j;
clrscr();
printf("Enter the names of cities...\n\n");
for(i=0;i<5;i++)
scanf("%s",&city[i]);
printf("Sorted list of cities...\n\n");
for(i=65;i<122;i++)
{
for(j=0;j<5;j++)
{
if(city[j][0]==i)
printf("\n%s",city[j]);
```



```
}  
}  
}
```

**OUTPUT:**

Enter the names of cities: Hyderabad Chennai Bombay Goa Vizag

Sorted list of cities:

Bombay

Chennai

Goa

Hyderabad

Vizag

RESULT: Thus the C program for sorting 5 city names in alphabetical order has been successfully executed and verified.

**Ex. No. 16 EXTRACTS THE PART OF A STRING**

**AIM: To write a program for extracting the part of a string.**

**ALGORITHM:**

Step1: Start the program.

Step2: Declare the character array s[30] and r[30].

Step3: Declare the integer variables i, j, m & n.

Step4: Get the input string using gets().

Step5: Get the value of m and n for extracting from the input string.

Step6: Initialize j=0.

Step7: Using a loop for(i=n-1;i<m+n-1;i++)

a) Assign r[j]=s[i];

b) Increment J by 1.

Step8: Print the extracted part of the string.

Step9: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<string.h>
void main()
{
char s[30],r[30];
int i,j,m,n;
clrscr();
printf("Enter a string: ");
gets(s);
printf("Enter the values of m & n: ");
scanf("%d%d",&m,&n);
j=0;
for(i=n-1;i<m+n-1;i++)
```

```
{  
r[j]=s[i];  
j++;  
}  
printf("The extracted part of string %s: ",r);  
getch();  
}
```

**OUTPUT:**

Enter a string: Gurunanak

Enter the values of m & n: 3 5

The extracted part of string: run

**RESULT:**

Thus the C program for extracting a part from the given string was executed and verified.

**Ex. No. 17 CALL BY VALUE**

**AIM: To write a program to increment the value of an argument using call by value.**

**ALGORITHM:**

Step1: Start the program.

Step2: Declare the integer variable x and a integer function incr()

Step3: Initialize x=7.

Step4: Pass the x value to the function incr(x).

a) Within the function increment the x value by 1.

b) Return the value.

Step5: Print the original value and incremented value of x.

Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>
#include<string.h>
main()
{ int x;
  int incr(int n);
  printf("***Call by Value***\n");
  x = 7;
  printf("Original value of x is: %d/n: ", x);
  printf("Value of incr(x) is: %d/n ", incr(x));
  printf("The value of x is: %d/n: ", x);
}
/* Function increments n */
int incr(int n)
{
n = n + 1;
```

```
return n;  
}
```

**OUTPUT:**

Original value of x is: 7

Value of incr(x) is : 8

The value of x is: 7

**RESULT:**

Thus the C program to increment the value of an argument using call by value was executed and verified.

### **Ex. No. 18 CALL BY REFERENCE**

**AIM: To write a program for swapping two values using call by reference method.**

#### **ALGORITHM:**

Step1: Start the program.

Step2: Assign the integer variable a=10 and b=20.

Step3: Call the swap() function.

Step4: Swap the values using pointer.

Step5: Print the original value and swapped value of a & b.

Step6: Stop the program.

#### **PROGRAM:**

```
#include<stdio.h>
#include<conio.h>
void swap( int *x, int *y )
{
int t ;
t = *x ;
*x = *y ;
*y = t ;
printf( "\nx = %d y = %d", *x,*y);
}
int main( )
{
int a = 10, b = 20 ;
swap ( &a, &b ) ;
printf ( "\na = %d b = %d", a, b ) ;
getch();
}
```

**OUTPUT:**

a=10 b=20

x=20 y=10

**RESULT:**

Thus the C program to swap two values using call by reference method was executed and verified.

**Ex. No. 19(a) STRUCTURE**

**AIM: To write a program for displaying student information by initializing structures.**

**ALGORITHM:**

Step1: Start the program.

Step2: Initialize a structure student with name as character array and roll number and age as integer.

Step3: In the main program create a object s1 for the structure student.

Step4: Using the object s1 print the student name, roll number and age.

Step6: Stop the program.

**PROGRAM:**

```
#include<stdio.h>

struct student
{
char name[10];
int rollno;
int age;
};

main()
{
static struct student s1;
clrscr();
printf("Enter the name, rollno & age");
scanf("%s%d%d\n",&s1.name,&s1.rollno,&s1.age);
printf("%s %d %d",s1.name,s1.rollno,s1.age);
getch();
}
```



**OUTPUT:**

Enter name, rollno & age

Ravi 11 25

Ravi 11 25

**RESULT:**

Thus the C program to display student information by initializing structures was executed and verified.

**Ex. No. 19(b) UNION**

**AIM: To write a program for implementing the concept of union data type.**

**ALGORITHM:**

Step1: Start the program.

Step2: Initialize a union Data with Str as character array, i as integer and f as float.

Step3: In the main program create a variable name data for the union Data.

Step4: Using the variable and member access operator print all the members of the union Data.

Step5: Stop the program.

**PROGRAM:**

```
#include <stdio.h>
#include <string.h>
union Data
{
int i;
float f;
char str[20];
};
int main( )
{
union Data data;
data.i = 10;
printf( "data.i : %d\n", data.i);
data.f = 220.5;
printf( "data.f : %f\n", data.f);
strcpy( data.str, "C Programming");
```

```
printf( "data.str : %s\n", data.str);  
return 0;  
}
```

OUTPUT:

data.i : 10

data.f : 220.500000

data.str : C Programming

**RESULT:**

Thus the C program to implement the concept of union data type was executed and verified.

**Ex. No. 20 ACCESS THE VALUE OF VARIABLES USING POINTER**

**AIM: To write a program for accessing the value of variables using pointer.**

**ALGORITHM:**

Step1: Start the program.

Step2: Declare integer as a, b, c and two pointer variables \*p1 & \*p2.

Step3: Intialize a=12 and b=4.

Step4: Assign the a & b values to the pointer variables p1 & p2.

Step5: Perform arithmetic operations.

Step6: Print the adderss of a & b and print the a, b, c, x & y values.

Step7: Stop the program.

**PROGRAM:**

```
#include<stdio.h>

main()
{
int a,b,*p1,*p2,x,y,z;
clrscr();
a=12,b=4;
p1=&a; p2=&b;
x=*p1**p2-6;
y=(4-*p2)**p1+10;
printf("Address of a=%d\n",p1);
printf("Address of b=%d\n",p2);
printf("a=%d,b=%d\n",a,b);
printf("x=%d,y=%d\n",x,y);
*p2=*p2+3; *p1=*p2-5;
z=*p1**p2-6;
printf("a=%d,b=%d\n",a,b);
```

```
printf("z=%d\n", z);  
getch();  
}
```

**OUTPUT:** Address of a = 65543

Address of b = 64455

a = 12 b = 4

x = y =

z=42

**RESULT:**

Thus the C program to access the value of variables using pointer was executed and

verified.

**Ex. No. 21 PRINT THE ELEMENT OF ARRAY USING POINTERS**

**AIM:**To write a program for printing the element of array using pointers.

**ALGORITHM:**

Step1: Start the program.

Step2: Declare integer array a[5] and a pointer variable \*p=&a[0]

Step3: Intialize i as integer.

Step4: Using the for loop for(i=0;i<5;i++)

Step5: prtint the value of \*(p+i).

Step6: Then using the for loop for(i=0;i<5;i++)

Stop7: Print the value of (p+1).

Step7: Stop the program.

**PROGRAM:**

```
#include<stdio.h>

main()
{
int a[5]={5,4,6,8,9};
int *p=&a[0];
int i;
clrscr();
for(i=0;i<5;i++)
printf("%d",*(p+i));
for(i=0;i<5;i++)
printf(" %u\n", (p+i));
getch();
}
```

**OUTPUT:**

1 2 3 4 5

1 2 3 4 5

**RESULT:**

Thus the C program to print the element of array using pointers was executed and verified.

**Ex. No. 22 PRINT THE ELEMENTS OF A STRUCTURE USING POINTERS**

**AIM: To write a program printing the elements of a structure using pointers.**

**ALGORITHM:**

Step1: Start the program.

step2: Take a character array name, a number and price in structure

step3: In main take a struct variable product and a pointer

Step4: Using a loop for(\*ptr=product;ptr<product+3;ptr++)

Step5: Read the value by using array operator

ptr->name,ptr->no,ptr->price

step6: Display name,no,price.

Step7: Stop the program.

**PROGRAM:**

```
#include<stdio.h>

struct invest
{
char name[20];
int number;
float price;
};

main()
{
struct invest product[3],*ptr;

clrscr();

printf("input\n\n");

for(*ptr=product[3];ptr<product+3;ptr++)

scanf("%s%d%f",&ptr->name,&ptr->number,&ptr->price);

printf("\nResult: \n\n");
```



```
ptr=product;
while(ptr<product+3)
{
printf("%20s%5d%10.2f\n",ptr->name,ptr->number,ptr->price);
ptr++;
}
getch();
}
```

**OUTPUT:**

```
Raja
11
120
```

**Result:**

```
Raja
11
120
```

**RESULT:**

Thus the C program to print the elements of a structure using pointers was executed and verified.

**Ex. No. 23 DISPLAY COLLEGE ADDRESS USING STRUCTURES AND POINTERS**

**AIM: To write a program for displaying college address using structures and pointers.**

**ALGORITHM:**

Step1: Start the program.

Step2: Take name, location and city inside the college address structure.

Step3: Enter the required data.

Step4: Print the result.

Step5: Stop the program.

**PROGRAM:**

```
#include<stdio.h>

struct collegeaddress

{

char name[20],location[20],city[20];

};

main()

{

struct collegeaddress add,*ptr;

p=&add;

p->name={"oseven"};

p->location={"America"};

p->city={"Florida"};

printf("%s%s%s",p->name,p->location,p->city);

}
```

**OUTPUT:**

oseven America Florida

**RESULT:**

Thus the C program to display college address using structures and pointers was executed and verified.

**Ex. No. 24 PASS STRUCTURE AS ARGUMENT TO FUNCTION**

**AIM: To write a program for passing structure as argument to function and calculate total marks of 5 subjects.**

**ALGORITHM:**

Step1: Start the program.

Step2: Inside the structure ex2 declare 6 integers.

Step3: Declare structure ex2 as s1.

Step4: Declare structure ex2 as s2,ex2 as fun().

Step5: Display the message as enter the marks.

Step6: Take value of the subjects from the user.

Step7: Store the return value in s2.total.

Step8: Print the value of s2.total.

Step9: Stop the program.

**PROGRAM:**

```
#include<stdio.h>

struct ex2
{
int m1,m2,m3,m4,m5,total;
};

main()
{
struct ex2 s1;
struct ex2 s2;
struct ex2 fun();
printf("enter the marks");
scanf("%d%d%d%d%d",&s1.m1,&s1.m2,&s1.m3,&s1.m4,&s1.m5);
s2=fun(s3);
printf("%d",s1.total);
```

```
}  
  
struct ex2 fun(s3)  
  
struct ex2 s3;  
  
{  
  
s3.total=s3.m1+s3.m2+s3.m3+s3.m4+s3.m5;  
  
return(s3);  
  
}
```

**OUTPUT:**

Enter the marks

10 20 30 40 50

150

**RESULT:**

Thus the C program to pass structure as argument to function and calculate total marks of 5 subjects was executed and verified.

**Ex. No. 25 WRITE INTEGER DATA INTO FILE AND READ IT FROM FILE**

**AIM: To write a program for writing integer data into file and read it from file.**

**ALGORITHM:**

Step1: Start the program.

Step2: Initialize integer num.

Step3: Declare FILE \*f2.

Step4: Open the file f2 using fopen() in write mode.

Step5: Get the integer from user and write it into the file using putw().

Step6: Close the file.

Step7: Open the file f2 using fopen() in read mode.

Step8: Read the integer using getw().

Step9: Print the integer.

Step10: Close the file.

Step11: Stop the program.

**PROGRAM:**

```
#include<stdio.h>

main()

{

int num;

FILE *f2;

f2=fopen("data.int", "w");

scanf ("%d", &num);

putw(num, f2);

fclose(f2);

f2=fopen("data.int", "r");

num=getw(f2);

printf("%d", num);
```

```
fclose(f2);
```

```
}
```

OUTPUT:

12

12

**RESULT:**

Thus the C program to write integer data into file and read it from file was executed and verified.