

4. Simple Menu Driven Calculator Program Using Switch Statement

Aim:

To write a C Program to write a simple menu driven calculator program using switch statement.

Algorithm:

Step 1: Start the program

Step 2: Read the two variable num1 and num2

Step 3: Display menu

Step 4: Enter the option code

Step 5: Evaluate option code with case statements

Step 5.1: case 1 ans1=num1+num2, print ans1. goto step 7

Step 5.2: case 2 ans1=num1-num2, print ans1. goto step 7

Step 5.3: case 3 ans1=num1*num2, print ans1. goto step 7

Step 5.4: case 4 ans2=(float)num1/num2, print ans2. goto step 7

Step 6: Entered case option is invalid code the print "Wrong Choice"

Step 7: Stop the program

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int num1,num2,ans1,choice;
    float ans2;
    clrscr();
    printf(" \n enter two numbers");
    scanf("%d%d",&num1,&num2);
    printf(" \n enter your choice \n 1.Addition \n 2.Subtraction \n 3.Multiplication \n
                                                4.Division \n");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1:
            ans1=num1+num2;
            printf("Addition =%d",ans1);
            break;
        case 2:
```

```

        ans1=num1-num2;
        printf("Subtraction =%d",ans1);
        break;
    case 3:
        ans1=num1*num2;
        printf("Multiplication =%d",ans1);
        break;
    case 4:
        ans2=(float)num1/num2;
        printf("Division =%f",ans2);
        break;
    default:
        printf("wrong choice");
        break;
    }
    getch();
}

```

/*OUTPUT FOR ADDITION:

enter two number

55

66

enter your choice

1.Addition

2.Subtraction

3.Multiplication

4.Division

1

Addition=121*/

/*OUTPUT FOR SUBTRACTION:

enter two number

67

66

enter your choice

1.Addition

2.Subtraction

3.Multiplication

4.Division

2

Subtraction=1*/

/*OUTPUT FOR MULTIPLICATION:

enter two number

4

2

enter your choice

1.Addition

2.Subtraction

3.Multiplication

4.Division

3

Multiplication=8 */

/*OUTPUT FOR DIVISION:

enter two number

55

5

enter your choice

1.Addition

2.Subtraction

3.Multiplication

4.Division

4

Division =11 */

/*OUTPUT FOR WRONG CHOICE:

enter two number

55

66

enter your choice

1.Addition

2.Subtraction

3.Multiplication

4.Division

5

wrong choice

Result:

Thus the program to create simple menu driven calculator using switch has been successfully executed.

5. Prime or Not

Aim:

To create a C program for checking and print of given number is prime or not.

Algorithm:

- Step 1: Start the program
- Step 2: Read the value of n
- Step 3: Assign i=2
- Step 4: Repeat the following steps until $i \leq n/2$
 - Step 4.1: If $n \% i = 0$ Then Print "Not Prime" and Exit.
 - Step 4.2: $i = i + 1$
- Step 5: if $i = n$ then Print "Prime Number"
- Step 6: Stop the program

Program:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int n, i, flag=0;
    clrscr();
    printf("\nEnter a positive integer: ");
    scanf("%d",&n);
    for(i=2;i<=n/2;i++)
    {
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }
}
```

```
if (flag==0)
    printf("\n%d is a prime number",n);
else
    printf("\n%d is not a prime number",n);
}
```

OUTPUT

Enter a positive integer: 5
5 is a prime number

Result:

Thus the program to find whether the given number is prime or not has been executed successfully.

6. Fibonacci Number

Aim:

To create a C program to generate the Fibonacci series for the given number.

Algorithm:

- Step 1: Start the program
- Step 2: Read the value of n
- Step 3: Assign f1 = 0, f2=1 and f=0
- Step 4: Print f1,f2
- Step 5: f=f1+f2
- Step 6: Repeat the following steps until f<n
 - Step 6.1 f = f1+f2
 - Step 6.2 f1 = f2
 - Step 6.3 f2 = f
 - Step 6.4 Print f
- Step 7: Stop the program

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int f,f1=0,f2=1,i;
```

```

clrscr();
printf("Enter the limit:");
scanf("%d",&n);
printf("\n%d\n%d",f1,f2);
for(i=3;i<=n;i++)
{
    f=f1+f2;
    printf("\n%d",f);
    f1=f2;
    f2=f;
}
getch();
}

```

OUTPUT

```

Enter the limit: 5
0
1
1
2
3

```

Result:

Thus the C program to generate the Fibonacci series for the given number has been executed successfully.

7. SUM OF DIGITS, REVERSE, PALINDROME

Aim:-

To write a C program to find the sum & reverse of digits and Check is Palindrome or not

Algorithm:-

STEP 1: Start the program.

STEP 2: Enter the number.

STEP 3: Set a loop upto the number is not equal to zero .

$rem \leftarrow num \% 10$

$sum \leftarrow sum + rem$

$rnum \leftarrow rnum * 10 + rem$

$num \leftarrow num / 10$

STEP 4: After the end of the loop print the sum and reverse no of the digit.

STEP 5: Find whether the reverse no is equal then is palindrome or not

STEP 6: Stop the Program.

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    unsigned long int a, num, sum=0,rnum=0,rem;
    clrscr();
    printf("\n Enter the No:");
    scanf("%ld",&num);
    a=num;
    while(num!=0)
    {
        rem=num%10;
        sum=sum+rem;
        rnum=rnum*10+rem;
        num=num/10;
    }
    printf("\n The Sum of Digits %ldis=%ld\n",a,sum);
    printf("\n The Reverse %ld is=%ld\n",a,rnum);
    if(a==rnum)
        printf("\n The Given number is a Palindrome");
    else
        printf("\n The Given number is not a Palindrome");
    getch();
}
```

Output:-

```
Enter the No:12345
The Sum of Digits 12345is=15
The Reverse 12345 is=54321
The Given number is not a Palindrome
```

Result:-

Thus the C program to find the sum & reverse of digits and check whether the number is Palindrome or Not is Verified successfully.

8. Roots of quadratic equation

Aim

To write a C program to find the roots of quadratic equation.

Algorithm

Step1: Start the program.

Step2: Read the value of a, b, c

Step3: Calculate $d = b^2 - 4ac$

Step4: If $d > 0$ then find the roots r_1 & r_2 by using the formula

$$r_1 = \frac{-b + \sqrt{d}}{2a}$$

$$r_2 = \frac{-b - \sqrt{d}}{2a}$$

Step5: Else if $d = 0$ then find the roots r by using the formula

$$r = \frac{-b}{2a}$$

Step6: Else, find the imaginary roots r_1 & r_2 by using the formula

$$r_1 = \frac{-b + \sqrt{d}}{2a}$$

$$r_2 = \frac{-b - \sqrt{d}}{2a}$$

Step7: Display the roots

Step8: Stop the program

Program

```
#include <stdio.h>
#include <math.h> /* This is needed to use sqrt() function.*/
int main()
{
    float a, b, c, determinant, r1, r2, real, imag;
    printf("Enter coefficients a, b and c: ");
    scanf("%f%f%f", &a, &b, &c);
    determinant = b*b - 4*a*c;
    if (determinant > 0)
    {
        r1 = (-b + sqrt(determinant)) / (2*a);
        r2 = (-b - sqrt(determinant)) / (2*a);
        printf("Roots are: %.2f and %.2f", r1, r2);
    }
    else if (determinant == 0)
    {
```



```

    r1 = r2 = -b/(2*a);
printf("Roots are: %.2f and %.2f", r1, r2);
}
else
{
real= -b/(2*a);
imag = sqrt(-determinant)/(2*a);
printf("Roots are: %.2f+%.2fi and %.2f-%.2fi", real, imag, real, imag);
}
return 0;
}

```

Output 1

Enter coefficients a, b and c: 2.3
4
5.6
Roots are: -0.87+1.30i and -0.87-1.30i

Output 2

Enter coefficients a, b and c: 4
1
0
Roots are: 0.00 and -0.25

Result

Thus the program for finding the roots of the quadratic equation is verified and executed successfully.

9. Sorting

Aim:

To create a program for sorting the number in an array ascending and descending order.

Algorithm:

Step 1: Start the program
Step 2: Read the value of n.

Step 3: Read the value of n array elements a[0],a[1]...a[n-1] using for loop.

Step 4: For i=0 to n

For j=i+1 to n

If a[i] > a[j] then assign temp=a[i], a[i]=a[j], a[j]=temp.

Step 5: For i=0 to n

Print the value of a[i] to print the numbers in ascending order

Step 6: For i=n-1 to 0

Print the value of a[i] to print the numbers in descending order

Step 7: Stop the program

Program:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    int a[10],n,i,j,temp;
```

```
    clrscr();
```

```
    printf("Enter the no of values");
```

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

```
    printf("Enter the elements one by one\n");
```

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

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

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

```
    {
```

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

```
        {
```

```
            if(a[i]>a[j])
```

```
            {
```

```
                temp=a[i];
```

```
                a[i]=a[j];
```

```
                a[j]=temp;
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("\n Ascending order");
```

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

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

```
    printf("\n Descending order");
```

```

        for(i=n-1;i>=0;i--)
            printf("\t%d",a[i]);
        getch();
    }

```

OUTPUT

Enter the no of values 5

Enter the elements one by one

23

12

54

76

90

Ascending order 12 23 54 76 90

Descending order 90 76 54 23 12

RESULT:

Thus the program to sort the number in ascending and descending order has been executed.

10. MATRIX ADDITION

Aim:

To create a C program for perform the matrix addition.

Algorithm:

Step 1: Start the program

Step 2: Read the number of rows and columns of two matrix say row1, col1, row2, col2.

Step 3: If row1 = col1 and row2 = col2 goto Step 5

Step 4: Print "Addition is not possible" goto Step 9

Step 5: For i=0 To row1-1

 For j=0 To col1-1

 Read the value of a[i][j]

Step 6: For i=0 To row2-1

 For j=0 To col2-1

 Read the value of b[i][j]

Step 7: For i=0 To row1-1

For j=0 To col1-1

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

Step 8: For i=0 To row1-1

For j=0 To col1-1

Print the value of c[i][j]

Step 9: Stop the program

Program:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    int a[5][5],b[5][5],c[5][5];
```

```
    int row1,row2,col1,col2,i,j,k;
```

```
    clrscr();
```

```
    printf("Enter the row value of first matrix\n");
```

```
    scanf("%d",&row1);
```

```
    printf("Enter the column value of first matrix\n");
```

```
    scanf("%d",&col1);
```

```
    printf("Enter the row value of second matrix\n");
```

```
    scanf("%d",&row2);
```

```
    printf("Enter the column value of second matrix\n");
```

```
    scanf("%d",&col2);
```

```
    if((row1==row2)&&(col1==col2))
```

```
    {
```

```
        printf("Matrix can be added \n");
```

```
        printf("Enter the values of first matrix\n");
```

```
        for(i=1;i<=row1;i++)
```

```
        {
```

```
            for(j=1;j<=col1;j++)
```

```
            {
```

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

```
            }
```

```
        }
```

```
        printf("Enter the values of second matrix\n");
```

```
        for(i=1;i<=row2;i++)
```

```
        {
```

```
            for(j=1;j<=col2;j++)
```

```

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

    for(i=1;i<=row1;i++)
    {
        for(j=1;j<=col1;j++)
        {
            c[i][j]=a[i][j]+b[i][j];
        }
    }
    printf("Sum of the two matrix is\n");
    for(i=1;i<=row1;i++)
    {
        for(j=1;j<=col1;j++)
        {
            printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
}
else
printf("Addition cannot be perform");
getch();
}

```

OUTPUT

```

Enter the row value of first matrix      2
Enter the column value of first matrix   2
Enter the row value of second matrix     2
Enter the column value of second matrix  2
Matrix can be added
Enter the values of first matrix
      2      4
      5      6
Enter the values of first matrix
      3      1
2      8

```

Sum of the two matrix

5	5
7	14

RESULT:

Thus the program to perform matrix addition has been executed successfully.

11. MATRIX MULTIPLICATION

AIM:-

To write a C program to perform Matrix Multiplication.

ALGORITHM:-

STEP 1: Start the program.

STEP 2: Enter the row and column of the A matrix.

STEP 3: Enter the row and column of the B matrix.

STEP 4: Enter the elements of the A matrix.

STEP 5: Enter the elements of the B matrix.

STEP 6: Print the elements of the A matrix in matrix form.

STEP 7: Print the elements of the B matrix in matrix form.

STEP 8: Set a loop up to row.

STEP 9: Set a inner loop up to column.

STEP 10: Set another inner loop up to column.

STEP 11: Multiply the A and B matrix and store the element in the C matrix.

STEP 12: Print the resultant matrix.

STEP 13: Stop the program.

PROGRAM:-

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[25][25],b[25][25],c[25][25],i,j,k,r,s;
    int m,n;
    clrscr();
    printf(" Enter the row and columns of A matrix.....");
    scanf("%d%d",&m,&n);
```

```

printf("Enter the row and columns of B matrix,,,,,,,,,");
scanf("%d%d",&r,&s);
if(n!=r)
    printf("\n The matrix cannot multiplied");
    for(i=0;i<r;i++)
    {
        for(j=0;j<s;j++)
        {
            scanf("%d",&b[i][j]);
        }
    }
printf("\n the elements of A matrix");
for(i=0;i<m;i++)
{
    printf("\n");
    for(j=0;j<n;j++)
    {
        printf("\t%d",a[i][j]);
    }
}
printf("\n the elements of B matrix");
for(i=0;i<m;i++)
{
    printf("\n");
    for(j=0;j<n;j++)
    {
        printf("\t%d",b[i][j]);
    }
}

for(i=0;i<m;i++)
{
    for(j=0;j<s;j++)
    {
        c[i][j]=0;
        for(k=0;k<n;k++)
        {

```

```

        c[i][j]=c[i][j]+a[i][k]*b[k][j];
    }
}
}
printf("The multiplication of two matrixes");
for(i=0;i<m;i++)
{
    printf("\n");
    for(j=0;j<s;j++)
    {
        printf("\t%d",c[i][j]);
    }
}
getch();
}

```

OUTPUT:-

Enter the row and columns of A matrix.....3 3

Enter the row and columns of B matrix,,,,,,,,,3 3

Enter the elements of A matrix1 2 3 4 5 6 7 8 9

Enter the elements of B matrix1 2 3 4 5 6 7 8 9

the elements of A matrix

1	2	3
4	5	6
7	8	9

the elements of B matrix

1	2	3
4	5	6
7	8	9

The multiplication of two matrixes

30	36	42
66	81	96
102	126	150

RESULT:-

Thus the C program to perform Matrix Multiplication is created successfully.

12. Searching

Aim

To write a C program to search the element in a list using linear search.

Algorithm

Step1: Start the program

Step2: Read the value of n

Step3: Get n values using for loop

Step4: Get the search item

Step5: Compare the list of items with search item in sequential order. If search item is matched with any array item then print "Element is found" and exit from the program

Step6: If search item is not matched with the array item, then print "Item is not found"

Step7: Display the result

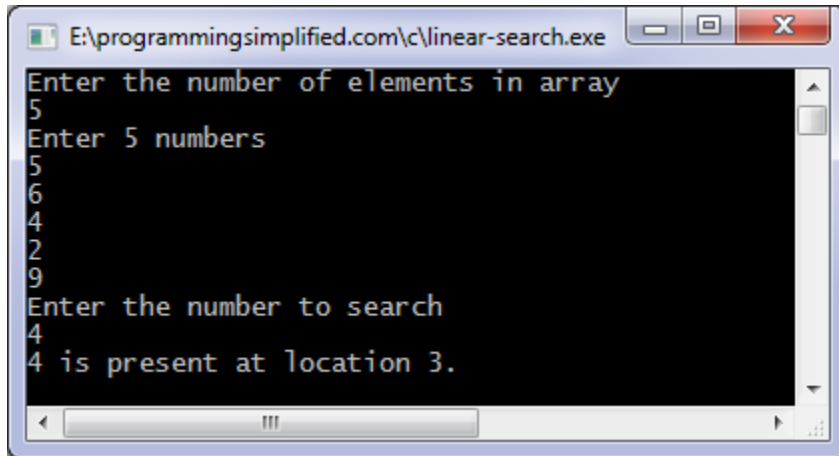
Step8: Stop the program

Program

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int array[100], search, c, n;
    clrscr();
    printf("Enter the number of elements in array\n");
    scanf("%d",&n);
    printf("Enter %d integer(s)\n", n);
    for (c = 0; c < n; c++)
    {
        scanf("%d", &array[c]);
    }
    printf("Enter the number to search\n");
    scanf("%d", &search);
    for (c = 0; c < n; c++)
    {
        if (array[c] == search)
        {
            printf("%d is present at location %d.\n", search, c+1);
            break;
        }
    }
}
```

```
}  
if (c == n)  
    printf("%d is not present in array.\n", search);  
    getch();  
}
```

Output



```
E:\programmingsimplified.com\c\linear-search.exe  
Enter the number of elements in array  
5  
Enter 5 numbers  
5  
6  
4  
2  
9  
Enter the number to search  
4  
4 is present at location 3.
```

Result

Thus the program for searching an element is verified and executed successfully.

13. String Sorting

Aim

To write a C program to sort the names in ascending order.

Algorithm

- Step1: Start the program
- Step2: Read the value of n
- Step3: Get n names using for loop
- Step4: For i=0 to n do
- Step5: For j=i+1 to n do
- Step6: If (strcmp(a[i],a[j])>0) then Swap a[i] & a[j] using strcpy function
- Step7: Repeat Step 4 & 5
- Step8: Display n names in sorting order
- Step9: Stop the program

Program

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    int i,j,n;
    char str[20][20],temp[20];
    clrscr();
    printf("\nEnter the no. of string to be sorted");
    scanf("%d",&n);
    printf("\nEnter n strings:\n");
    for(i=0;i<=n;i++)
        scanf("%s",str[i]);
    for(i=0;i<=n;i++)
    {
        for(j=i+1;j<=n;j++)
        {
            if(strcmp(str[i],str[j])>0)
            {
                strcpy(temp,str[i]);
                strcpy(str[i],str[j]);
                strcpy(str[j],temp);
            }
        }
    }
    printf("The sorted string\n");
    for(i=0;i<=n;i++)
        printf("%s",str[i]);
    getch();
}

```

Output

Enter the no. of string to be sorted:5

Enter n strings:

Balu
Vinoth
Priya
Sekar
Raju

The sorted string

Balu
Priya
Raju
Sekar

Vinoth

Result

Thus the program for sorting n names in ascending order is executed and verified successfully.

14. Factorial using Recursion

Aim

To write a C program to find the factorial using recursion.

Algorithm

- Step1: Start the program
- Step2: Declare the function factorial with return type int with one integer argument.
- Step3: Read the value of n in main() function
- Step4: Call the function factorial(n) from the main()
- Step5: If n=1 then return 1
- Step6: Otherwise, call the function factorial recursively by
factorial(i) = i * factorial (i-1)
- Step6: Return the result of the factorial(n) to the main procedure and display the result
- Step7: Stop the program

Program

```
#include <stdio.h>
#include <conio.h>
int factorial(int);
void main()
{
    int n;
    clrscr();
    printf("\n Enter n:");
    scanf("%d",&n);
    printf("Factorial is %d", factorial(n));
    getch();
}
int factorial(int i)
{
    if(i <= 1)
    {
        return 1;
```

```

    }
    return i * factorial(i - 1);
}

```

Output

Enter n: 5
Factorial is 120

Result

Thus the program for finding factorial using function is executed and verified successfully.

15. Parameter Passing -Call by Value

Aim

To write a program for swapping using call by value

Algorithm

- Step1: Start the program
- Step2: Declare the function swap() with two arguments
- Step3: Declare and Initialize the two variables a & b in main()
- Step4: Display the value of a & b
- Step5: Call the function swap() and pass the value of a & b as arguments
- Step6: Value of a & b is passed to x & y in the function swap
- Step7: Swap the value of x & y using temp
- Step8: Display the value of a & b from main()
- Step9: Stop the program

Program

```

#include <stdio.h>
void swap(int, int);           // function declaration
void main ()
{
    int a = 100, b=200;
    printf("Before swap, value of a : %d\n", a );
    printf("Before swap, value of b : %d\n", b );
    swap(a, b);                //Function Call
    printf("After swap, value of a : %d\n", a );
    printf("After swap, value of b : %d\n", b );
}
void swap(int x, int y)        //function definition
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}

```

Output

Before swap, value of a :100
Before swap, value of b :200
After swap, value of a :100
After swap, value of b :200

Result

Thus the program for swapping using call by value is executed and verified successfully.

16. Parameter Passing -Call by Reference

Aim

To write a program for swapping using Call by Reference

Algorithm

Step1: Start the program
Step2: Declare the function swap() with two pointers
Step3: Declare and Initialize the two variables a & b in main()
Step4: Display the value of a & b
Step5: Call the function swap() and pass the address of a & b as arguments
Step6: Address of a & b is passed to x & y in the function swap
Step7: Swap the value of x & y using temp
Step8: Display the value of a & b from main()
Step9: Stop the program

Program

```
#include <stdio.h>
void swap(int *, int *);           // function declaration
void main ()
{
    int a = 100, b=200;
    printf("Before swap, value of a : %d\n", a );
    printf("Before swap, value of b : %d\n", b );
    swap(&a, &b);                  //Function Call
    printf("After swap, value of a : %d\n", a );
    printf("After swap, value of b : %d\n", b );
}
void swap(int *x, int *y)         //function definition
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
```

Output

Before swap, value of a :100
Before swap, value of b :200
After swap, value of a :200
After swap, value of b :100

Result

Thus the program for swapping using call by value is executed and verified successfully.

17. Structures**Aim:**

To create a program to print the mark sheet of 'n' student using structures

Algorithm:

Step 1: Start the program
Step 2: Create the structure student with fields name, rollno, m1,m2,m3,total.
Step 3: Create the structure variable s1[10] for the structure student.
Step 4: Read the number of students num.
Step 5: Read the value of name, rollno, marks (m1,m2,m3) for the specified number of students using structure variable s1[i] for i= 0 to num.
Step 6: Calculate the total for each student using s1[i] for i=0 to num.
Step 7: For i=0 to num
 Print the students details such as name, rollno, marks and total for all students using s1[i]
Step 8: Stop the program

Program:

```
#include<stdio.h>
#include<conio.h>
struct student
{
    char name[10][10];
    int rollno,m1,m2,m3,total;
};
void main()
{
    int num,i,j;
    struct student s1[10];
    clrscr();
    printf("enter the number of students");
    scanf("%d",&num);
    for(i=0;i<num;i++)
    {
        printf("Enter the roll number\n");
        scanf("%d",&s1[i].rollno);
```

```

        printf("Enter the name \n");
        scanf("%s",&s1[i].name);
        printf("Enter the mark1\n");
        scanf("%d",&s1[i].m1);
        printf("Enter the mark2\n");
        scanf("%d",&s1[i].m2);
        printf("Enter the mark3\n");
        scanf("%d",&s1[i].m3);
        s1[i].total=s1[i].m1+s1[i].m2+s1[i].m3;
    }
    printf("The details of the mark list is as follows \n");
    printf("\n Rollno \t name \t mark1 \t mark2 \t mark3 \t total \n");
    for(i=0; i<num; i++)
    {
        printf("\n%d\t%s\t%d\t%d\t%d",s1[i].rollno,s1[i].name,s1[i].m1,s1[i].m2,s1[i].m3);
        s1[i].total=s1[i].m1+s1[i].m2+s1[i].m3;
        printf("\t%d",s1[i].total);
    }
    getch();
}

```

Output:

```

enter the number of stuents2
Enter the roll number
4561
Enter the name
lokesh
Enter the mark1  98
Enter the mark2  78
Enter the mark3  69
Enter the roll number
4562
Enter the name
mani
Enter the mark1  88
Enter the mark2  89
Enter the mark3  98
The details of the mark list is as follows

```

Rollno	name	mark1	mark2	mark3	total
4561	lokesh	98	78	69	245
4562	mani	88	89	98	275

Result:

The program to print the mark sheet of students using structure has been executed successfully.

18. Union

Aim:

To create a "C" program to demonstrate the union.

Algorithm:

- Step 1: Start the program
- Step 2: Create the union Data with fields of integer variable i, float variable f and character array variable str.
- Step 3: Create the union variable data for the union Data.
- Step 4: Print the size of union using sizeof() operator.
- Step 5: Assign the value to variable i and f using union variable data.
- Step 6: Print the value of i and f using union variable data.
- Step 7: Copy the string say "C Programming" to str variable using union variable data.
- Step 8: Print the string using union variable data.
- Step 9: Stop the program

Program:

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
union Data
{
    int i;
    float f;
    char str[20];
};
int main( )
{
    union Data d;
    clrscr();
    printf( "Memory size occupied by data : %d\n", sizeof(data));
    d.i = 10;
    d.f = 220.5;
    strcpy( data.str, "C Programming");
    printf( "%d\n%f\n%s", d.i, d.f, d.str);
    return 0;
}
```

Output:

Memory size occupied by data : 20
data.i : 10
data.f : 220.500000

data.str : C Programming

Result:

The program to demonstrate the union has been executed successfully.