BHARTIYA INSTITUTE OF ENGINEERING & TECHNOLOGY, SIKAR

LAB MANUAL

(PART II)

FIRST YEAR

(II SEMESTER)

Subject Code: 2FY3-24

COMPUTER PROGRAMMING LAB

Prepared By:
Mr. AMIT SINGH SHEKHAWAT
Head of Department
(CSE Dept.)

Prepared By:
Mr. PANKAJ GOTHWAL
Assistant Professor
(CSE Dept.)

Department of Computer Science & Engineering

BIET, Sikar

Website: www.bietsikar.ac.in

LIST OF EXPERIMENTS

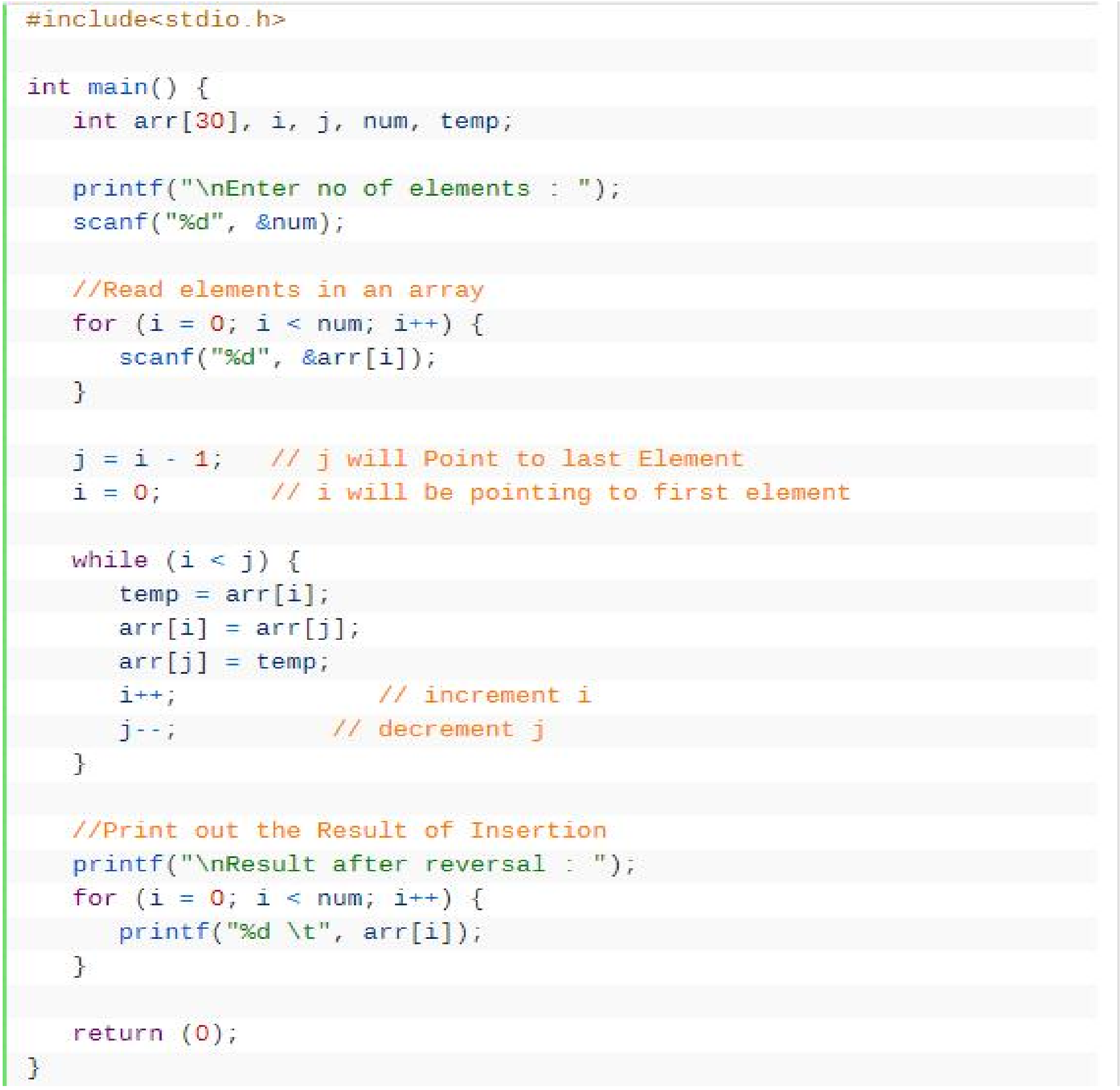
The programs shall be developed in C language related with the following concepts:

1. Input roll numbers of your friends in an array & print in reverse order.
2. Input names of your friends in an array & print in reverse order.
3. Input two matrices and output third matrix after performing add/subtract the corresponding elements.
4. Four programs using malloc, calloc, free & sscanf()/sprintf() functions.
5. Two programs using macro and online functions.
6. Two programs using structure & union.
7. Two programs using pointers.
8. Three programs belonging to file operations and multi-file handling.
9. Three programs belonging to graphics using C.

EXPERIMENT: 1

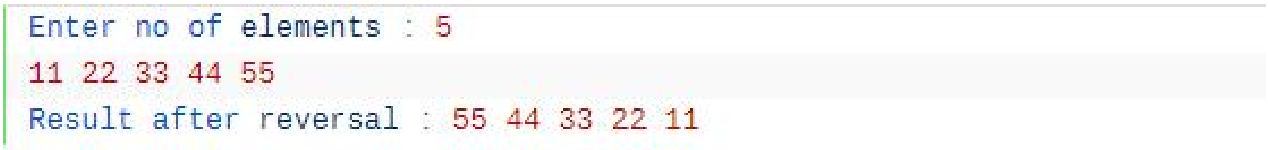
Aim: Program for Input roll numbers of your friends in an array & print in reverse order.

Code:



```
#include<stdio.h>

int main() {
   int arr[30], i, j, num, temp;

   printf("\nEnter no of elements : ");
   scanf("%d", &num);

   //Read elements in an array
   for (i = 0; i < num; i++) {
      scanf("%d", &arr[i]);
   }

   j = i - 1;   // j will Point to last Element
   i = 0;       // i will be pointing to first element

   while (i < j) {
      temp = arr[i];
      arr[i] = arr[j];
      arr[j] = temp;
      i++;             // increment i
      j--;          // decrement j
   }

   //Print out the Result of Insertion
   printf("\nResult after reversal : ");
   for (i = 0; i < num; i++) {
      printf("%d \t", arr[i]);
   }

   return (0);
}
```

Output:

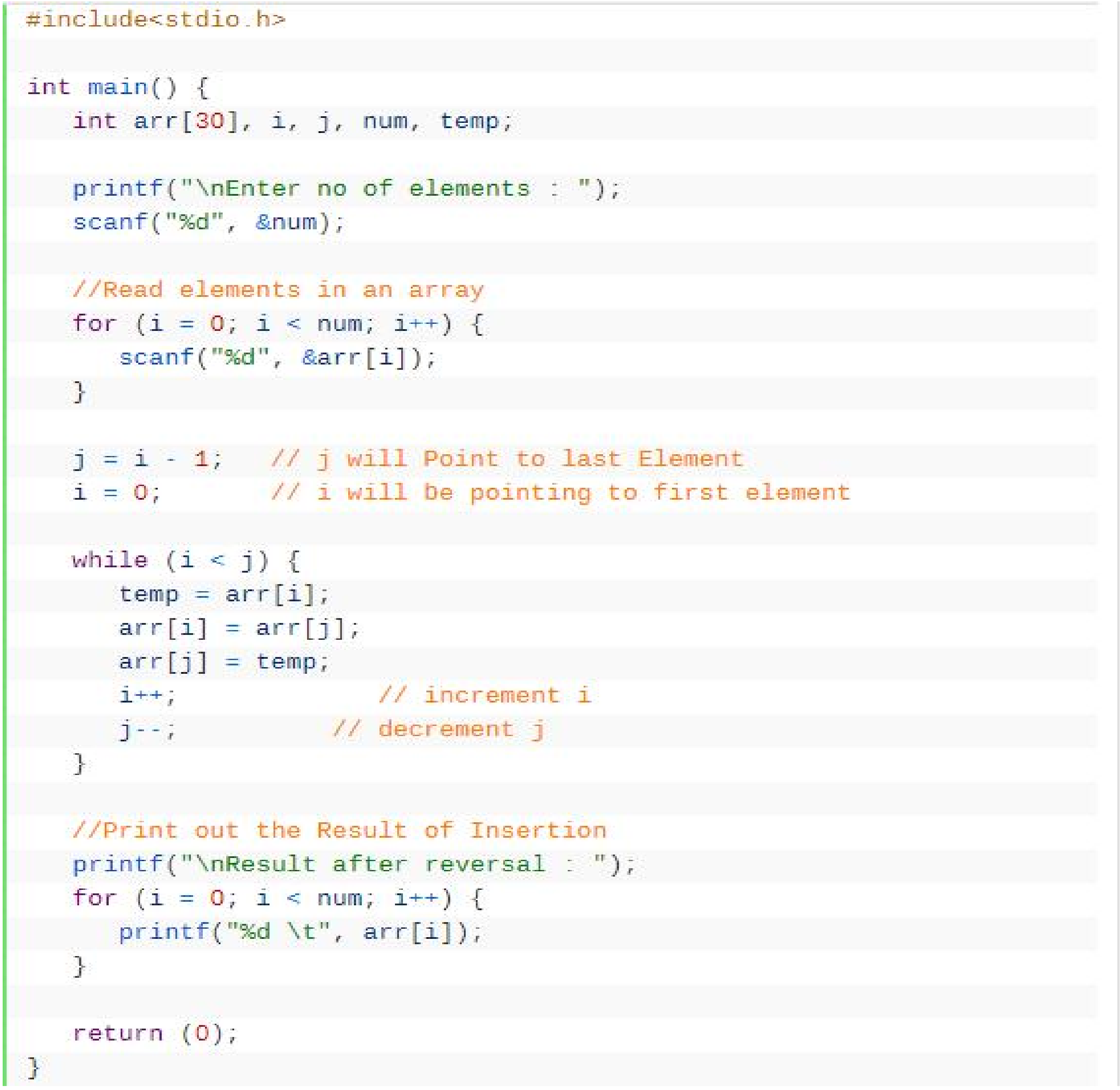


```
Enter no of elements : 5
11 22 33 44 55
Result after reversal : 55 44 33 22 11
```

EXPERIMENT: 2

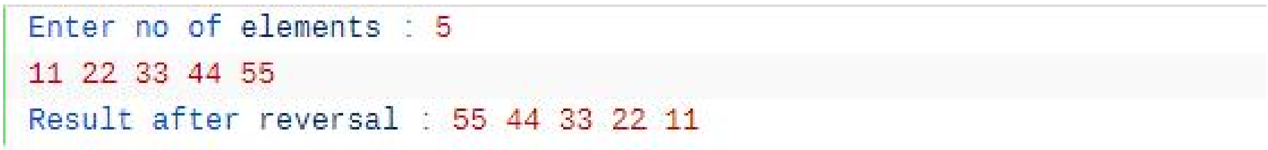
Aim: Program for Input names of your friends in an array & print in reverse order.

Code:



```
#include<stdio.h>

int main() {
   int arr[30], i, j, num, temp;

   printf("\nEnter no of elements : ");
   scanf("%d", &num);

   //Read elements in an array
   for (i = 0; i < num; i++) {
      scanf("%d", &arr[i]);
   }

   j = i - 1;   // j will Point to last Element
   i = 0;       // i will be pointing to first element

   while (i < j) {
      temp = arr[i];
      arr[i] = arr[j];
      arr[j] = temp;
      i++;             // increment i
      j--;          // decrement j
   }

   //Print out the Result of Insertion
   printf("\nResult after reversal : ");
   for (i = 0; i < num; i++) {
      printf("%d \t", arr[i]);
   }

   return (0);
}
```

Output:

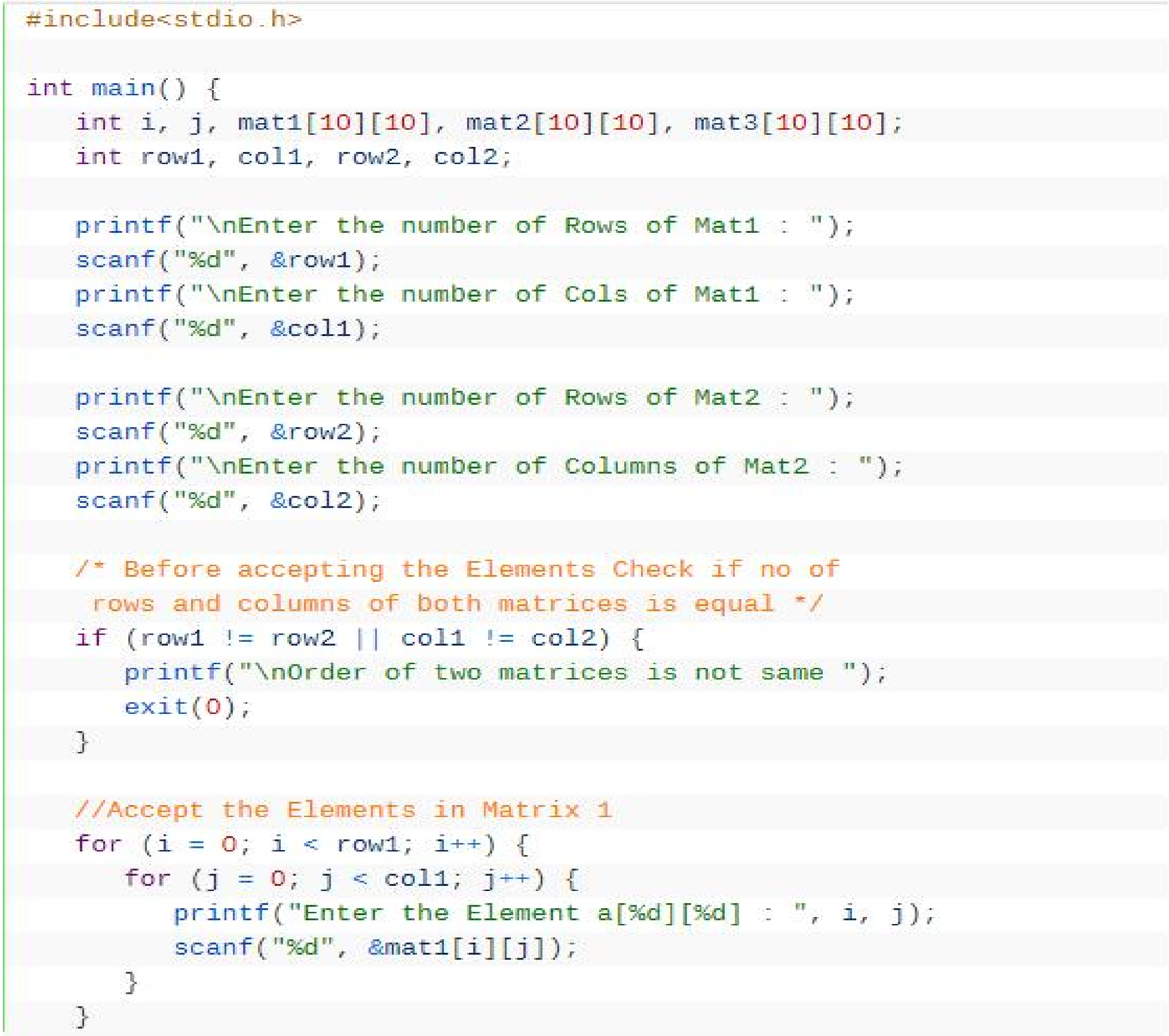


```
Enter no of elements : 5
11 22 33 44 55
Result after reversal : 55 44 33 22 11
```

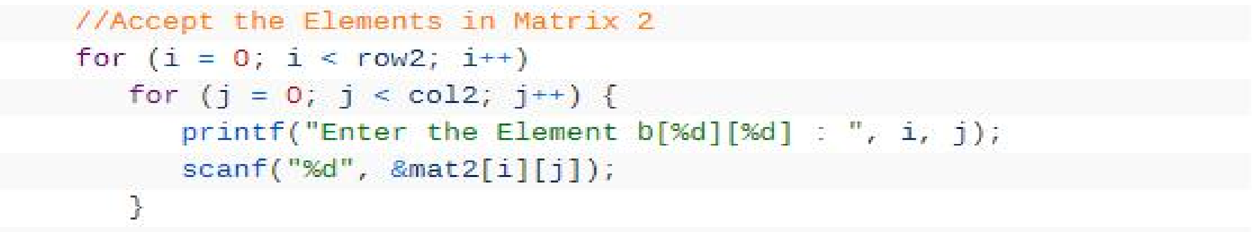
EXPERIMENT: 3

Aim: Program for Input two matrices and output third matrix after performing add/subtract the corresponding elements.

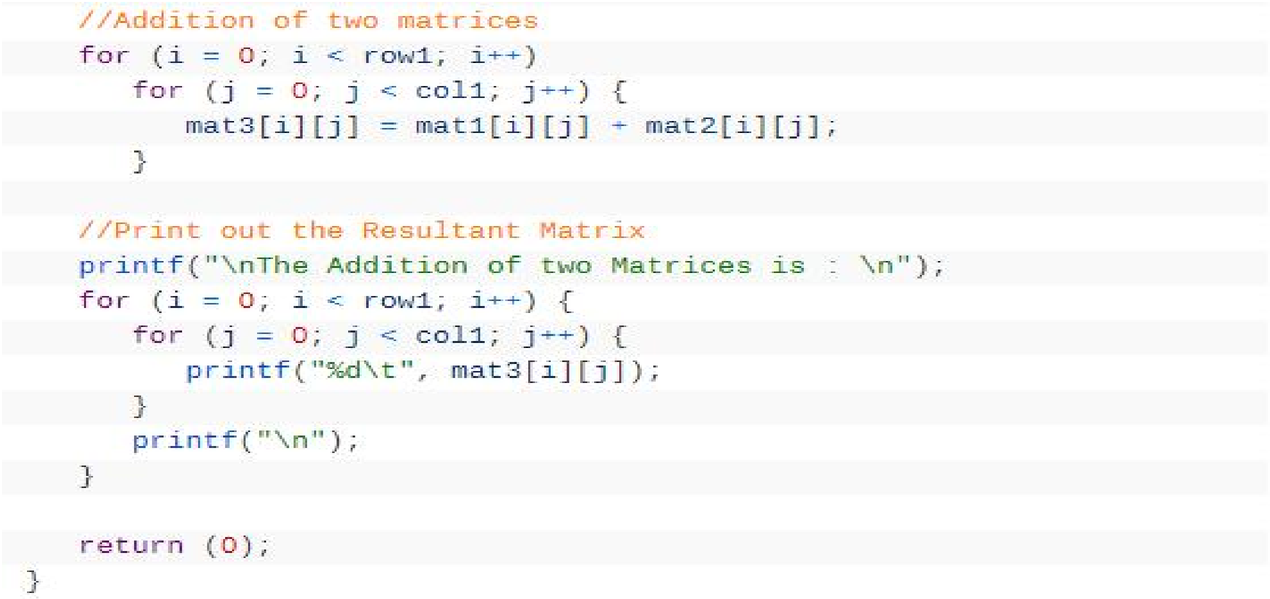
Code:



```
#include<stdio.h>

int main() {
   int i, j, mat1[10][10], mat2[10][10], mat3[10][10];
   int row1, col1, row2, col2;

   printf("\nEnter the number of Rows of Mat1 : ");
   scanf("%d", &row1);
   printf("\nEnter the number of Cols of Mat1 : ");
   scanf("%d", &col1);

   printf("\nEnter the number of Rows of Mat2 : ");
   scanf("%d", &row2);
   printf("\nEnter the number of Columns of Mat2 : ");
   scanf("%d", &col2);

   /* Before accepting the Elements Check if no of
    rows and columns of both matrices is equal */
   if (row1 != row2 || col1 != col2) {
      printf("\nOrder of two matrices is not same ");
      exit(0);
   }

   //Accept the Elements in Matrix 1
   for (i = 0; i < row1; i++) {
      for (j = 0; j < col1; j++) {
         printf("Enter the Element a[%d][%d] : ", i, j);
         scanf("%d", &mat1[i][j]);
      }
   }
```

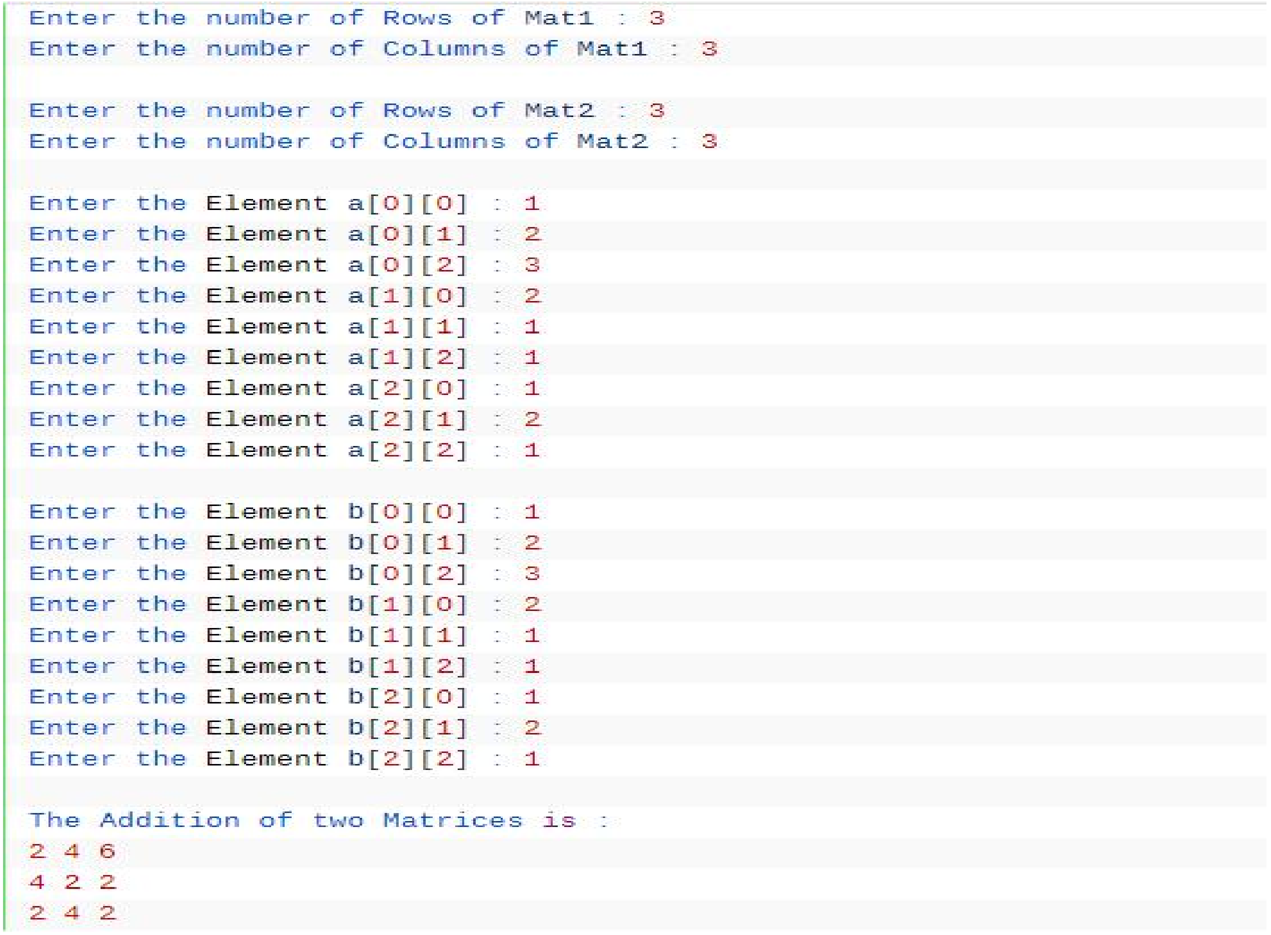


```
   //Accept the Elements in Matrix 2
   for (i = 0; i < row2; i++)
      for (j = 0; j < col2; j++) {
         printf("Enter the Element b[%d][%d] : ", i, j);
         scanf("%d", &mat2[i][j]);
      }
```



```
   //Addition of two matrices
   for (i = 0; i < row1; i++)
      for (j = 0; j < col1; j++) {
         mat3[i][j] = mat1[i][j] + mat2[i][j];
      }

   //Print out the Resultant Matrix
   printf("\nThe Addition of two Matrices is : \n");
   for (i = 0; i < row1; i++) {
      for (j = 0; j < col1; j++) {
         printf("%d\t", mat3[i][j]);
      }
      printf("\n");
   }

   return (0);
}
```

Output:

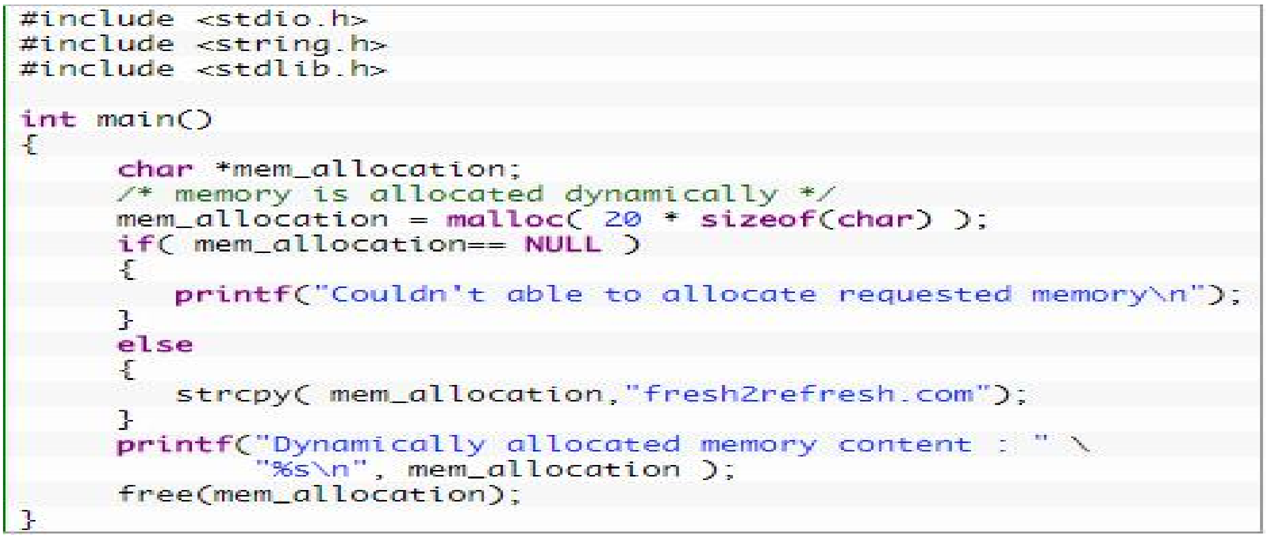


```
Enter the number of Rows of Mat1 : 3
Enter the number of Columns of Mat1 : 3

Enter the number of Rows of Mat2 : 3
Enter the number of Columns of Mat2 : 3

Enter the Element a[0][0] : 1
Enter the Element a[0][1] : 2
Enter the Element a[0][2] : 3
Enter the Element a[1][0] : 2
Enter the Element a[1][1] : 1
Enter the Element a[1][2] : 1
Enter the Element a[2][0] : 1
Enter the Element a[2][1] : 2
Enter the Element a[2][2] : 1

Enter the Element b[0][0] : 1
Enter the Element b[0][1] : 2
Enter the Element b[0][2] : 3
Enter the Element b[1][0] : 2
Enter the Element b[1][1] : 1
Enter the Element b[1][2] : 1
Enter the Element b[2][0] : 1
Enter the Element b[2][1] : 2
Enter the Element b[2][2] : 1

The Addition of two Matrices is :
2 4 6
4 2 2
2 4 2
```

EXPERIMENT: 4 (A)

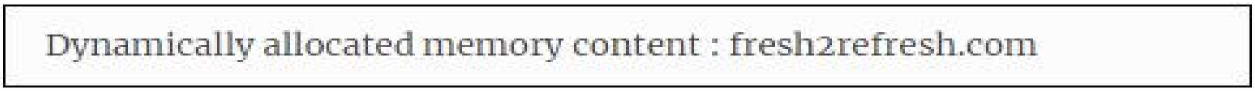
Aim: Program for implement the malloc () function in c programming.

Code:



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

int main()
{
     char *mem_allocation;
     /* memory is allocated dynamically */
     mem_allocation = malloc( 20 * sizeof(char) );
     if( mem_allocation== NULL )
     {
        printf("Couldn't able to allocate requested memory\n");
     }
     else
     {
        strcpy( mem_allocation,"fresh2refresh.com");
     }
     printf("Dynamically allocated memory content : " \
            "%s\n", mem_allocation );
     free(mem_allocation);
}
```

Output:

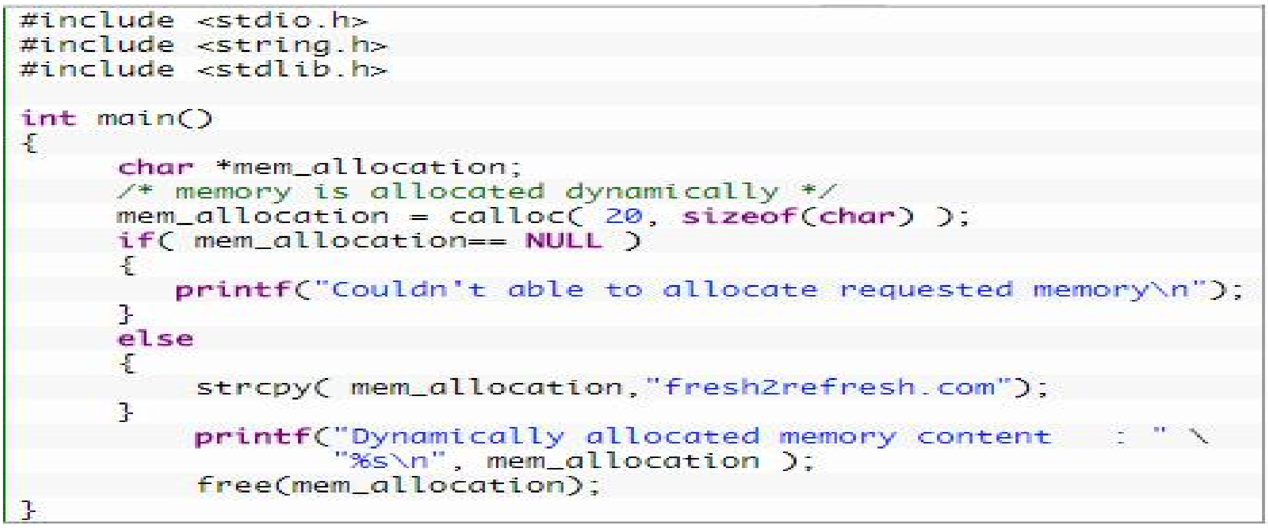


```
Dynamically allocated memory content : fresh2refresh.com
```

EXPERIMENT: 4 (B)

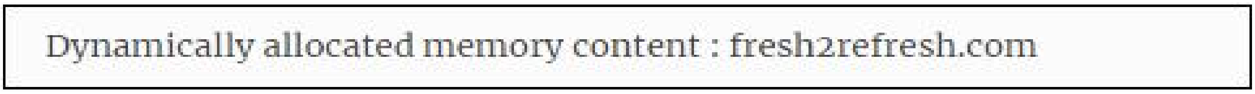
Aim: Program for implement the calloc () function in c programming.

Code:



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

int main()
{
     char *mem_allocation;
     /* memory is allocated dynamically */
     mem_allocation = calloc( 20, sizeof(char) );
     if( mem_allocation== NULL )
     {
        printf("Couldn't able to allocate requested memory\n");
     }
     else
     {
         strcpy( mem_allocation,"fresh2refresh.com");
     }
         printf("Dynamically allocated memory content   : " \
                "%s\n", mem_allocation );
         free(mem_allocation);
}
```

Output:

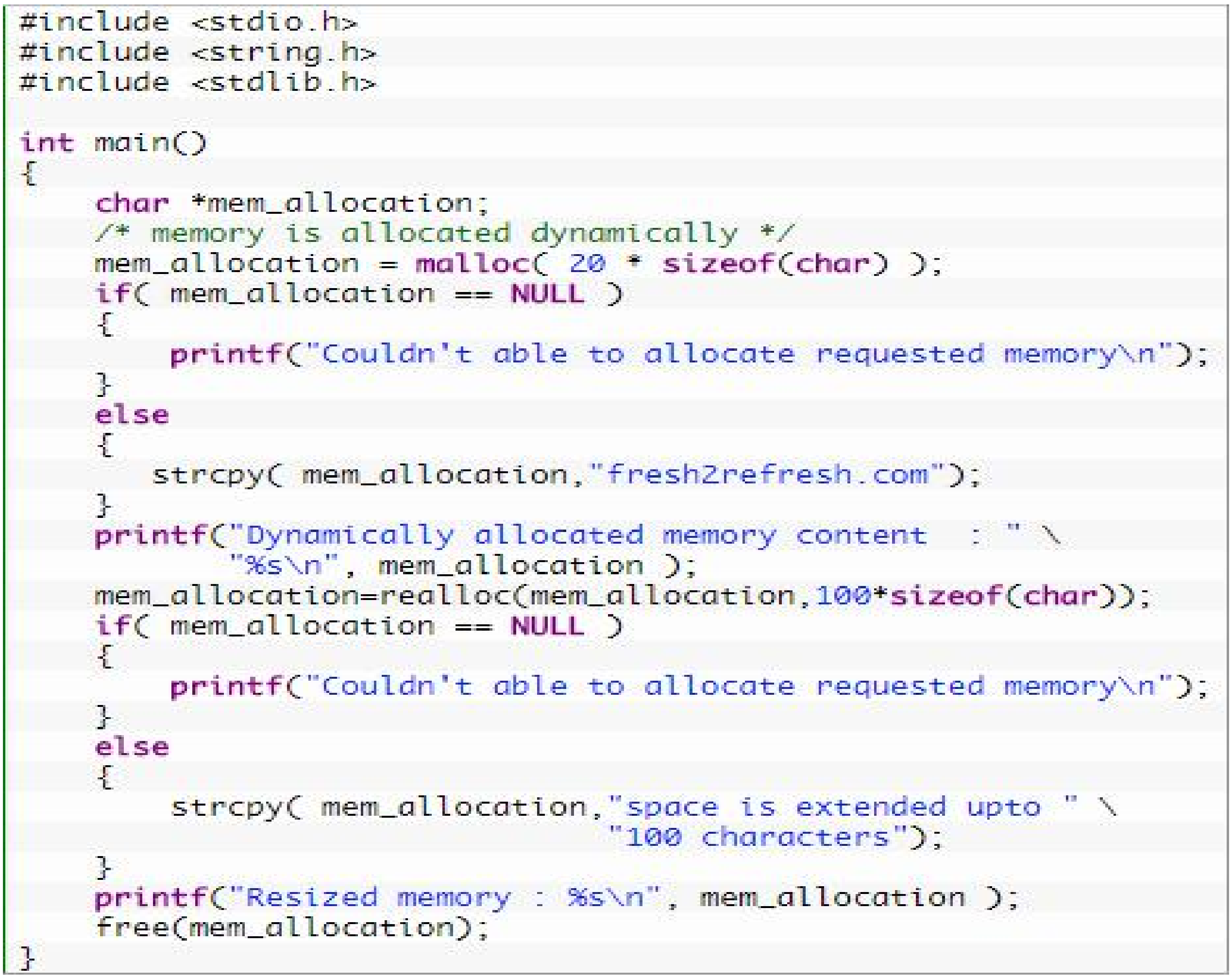


```
Dynamically allocated memory content : fresh2refresh.com
```

EXPERIMENT: 4 (C)

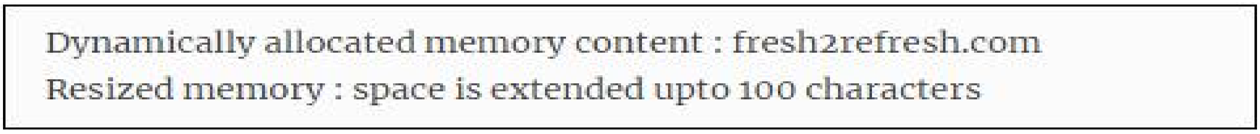
Aim: Program for implement the free () function in c programming.

Code:



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

int main()
{
    char *mem_allocation;
    /* memory is allocated dynamically */
    mem_allocation = malloc( 20 * sizeof(char) );
    if( mem_allocation == NULL )
    {
        printf("Couldn't able to allocate requested memory\n");
    }
    else
    {
       strcpy( mem_allocation,"fresh2refresh.com");
    }
    printf("Dynamically allocated memory content  : " \
           "%s\n", mem_allocation );
    mem_allocation=realloc(mem_allocation,100*sizeof(char));
    if( mem_allocation == NULL )
    {
        printf("Couldn't able to allocate requested memory\n");
    }
    else
    {
        strcpy( mem_allocation,"space is extended upto " \
                           "100 characters");
    }
    printf("Resized memory : %s\n", mem_allocation );
    free(mem_allocation);
}
```

Output:

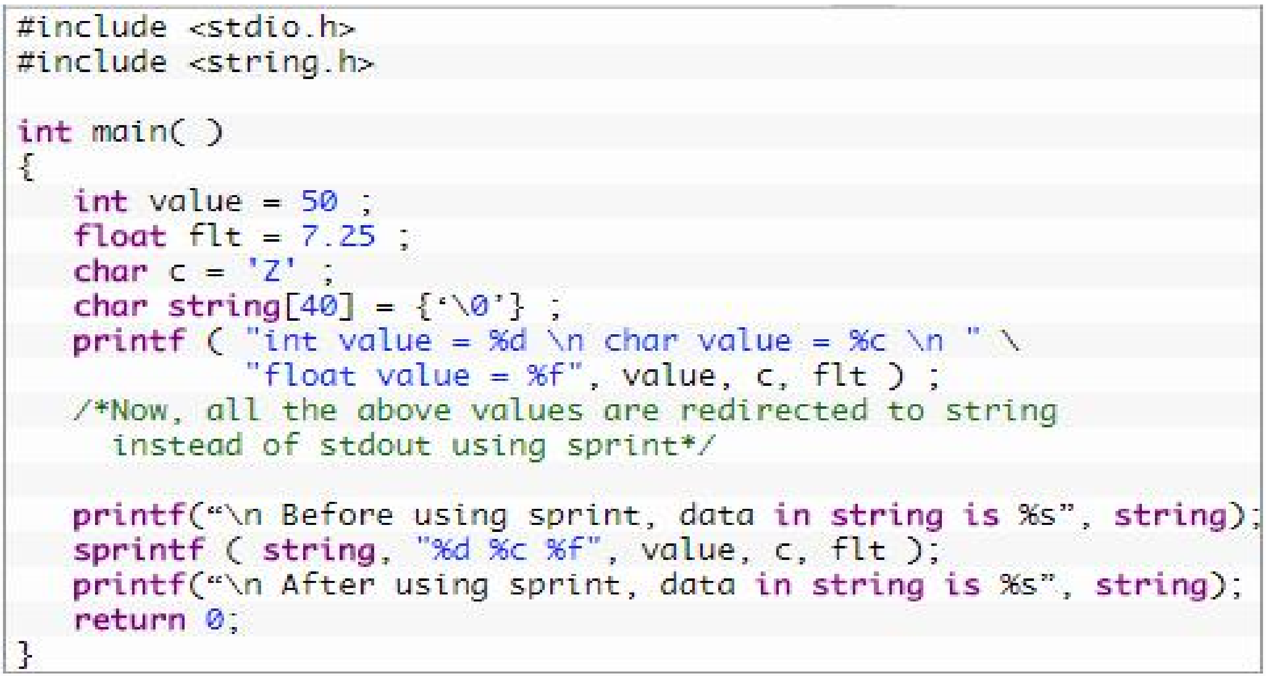


```
Dynamically allocated memory content : fresh2refresh.com
Resized memory : space is extended upto 100 characters
```

EXPERIMENT: 4 (D)

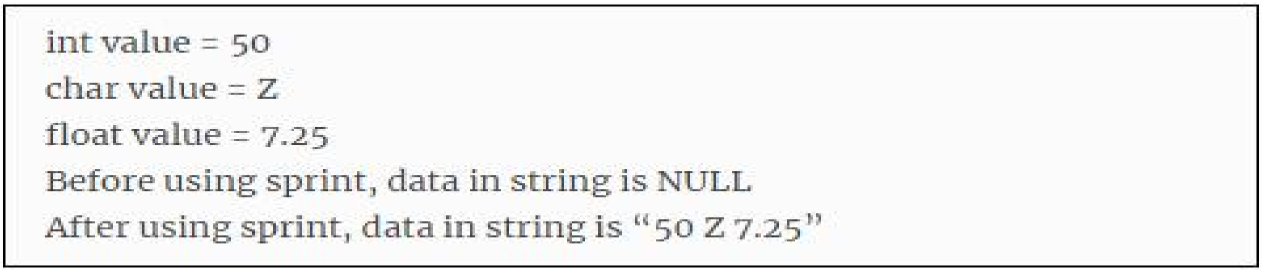
Aim: Program for implement the sprintf() and sscanf() function in c programming.

Code:



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

int main( )
{
   int value = 50 ;
   float flt = 7.25 ;
   char c = 'Z' ;
   char string[40] = {‘\0’} ;
   printf ( "int value = %d \n char value = %c \n " \
            "float value = %f", value, c, flt ) ;
   /*Now, all the above values are redirected to string
     instead of stdout using sprint*/

   printf(“\n Before using sprint, data in string is %s”, string);
   sprintf ( string, "%d %c %f", value, c, flt );
   printf(“\n After using sprint, data in string is %s”, string);
   return 0;
}
```

Output:

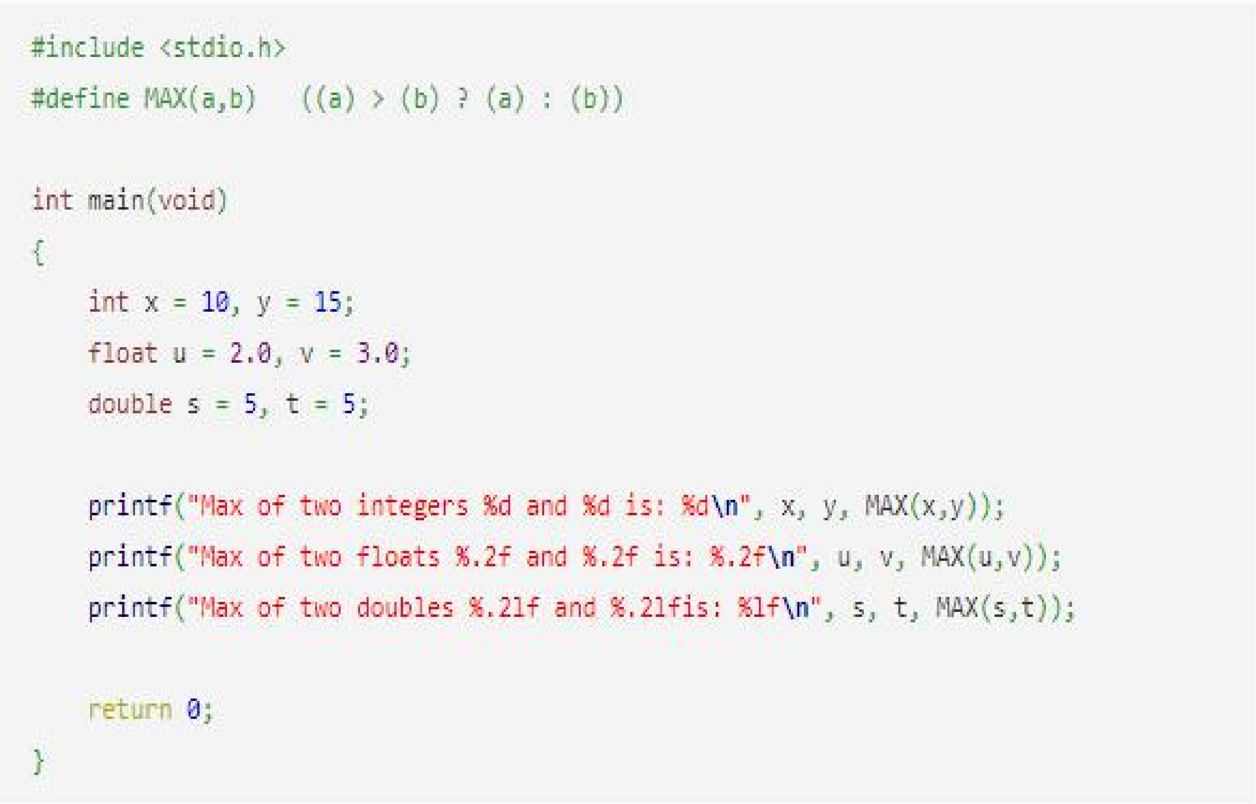


```
int value = 50
char value = Z
float value = 7.25
Before using sprint, data in string is NULL
After using sprint, data in string is “50 Z 7.25”
```

EXPERIMENT: 5 (A)

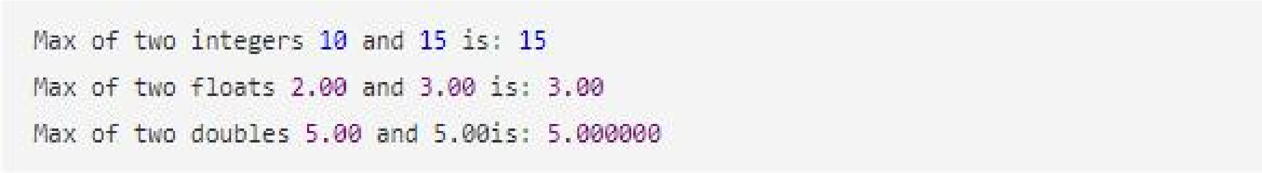
Aim: Program for implement the macro function in c programming.

Code:



```c
#include <stdio.h>
#define MAX(a,b)   ((a) > (b) ? (a) : (b))

int main(void)
{
    int x = 10, y = 15;
    float u = 2.0, v = 3.0;
    double s = 5, t = 5;

    printf("Max of two integers %d and %d is: %d\n", x, y, MAX(x,y));
    printf("Max of two floats %.2f and %.2f is: %.2f\n", u, v, MAX(u,v));
    printf("Max of two doubles %.2lf and %.2lfis: %lf\n", s, t, MAX(s,t));

    return 0;
}
```

Output:

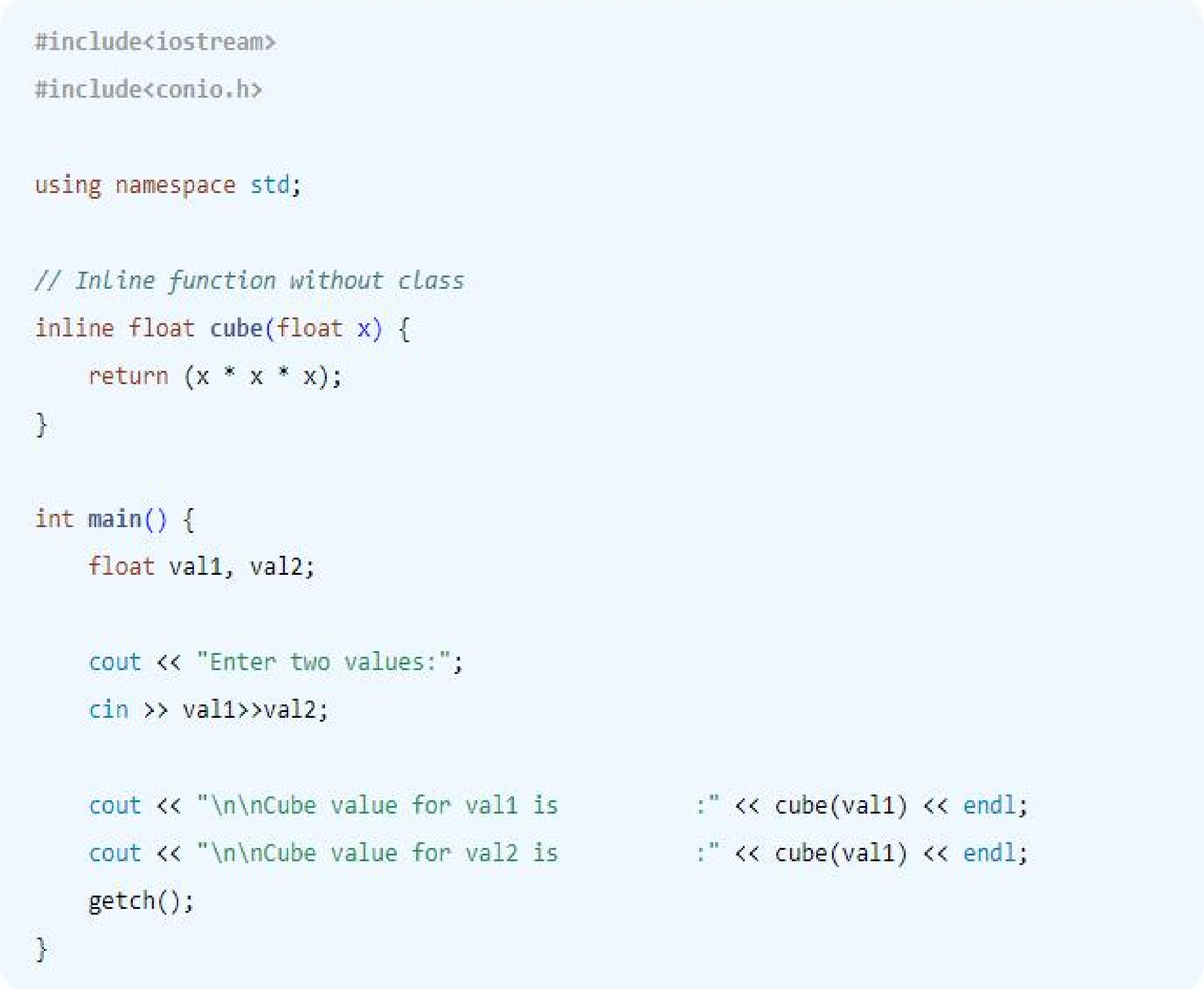


```
Max of two integers 10 and 15 is: 15
Max of two floats 2.00 and 3.00 is: 3.00
Max of two doubles 5.00 and 5.00is: 5.000000
```

EXPERIMENT: 5 (B)

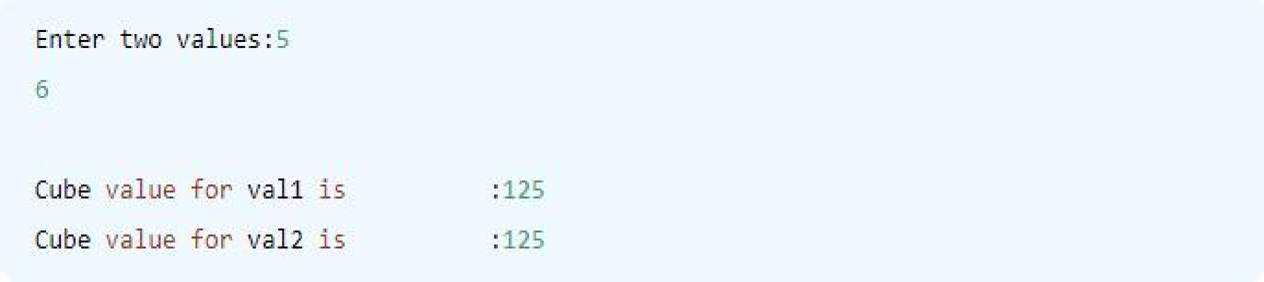
Aim: Program for implement the inline function in c programming.

Code:



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

using namespace std;

// Inline function without class
inline float cube(float x) {
    return (x * x * x);
}

int main() {
    float val1, val2;

    cout << "Enter two values:";
    cin >> val1>>val2;

    cout << "\n\nCube value for val1 is          :" << cube(val1) << endl;
    cout << "\n\nCube value for val2 is          :" << cube(val1) << endl;
    getch();
}
```

Output:

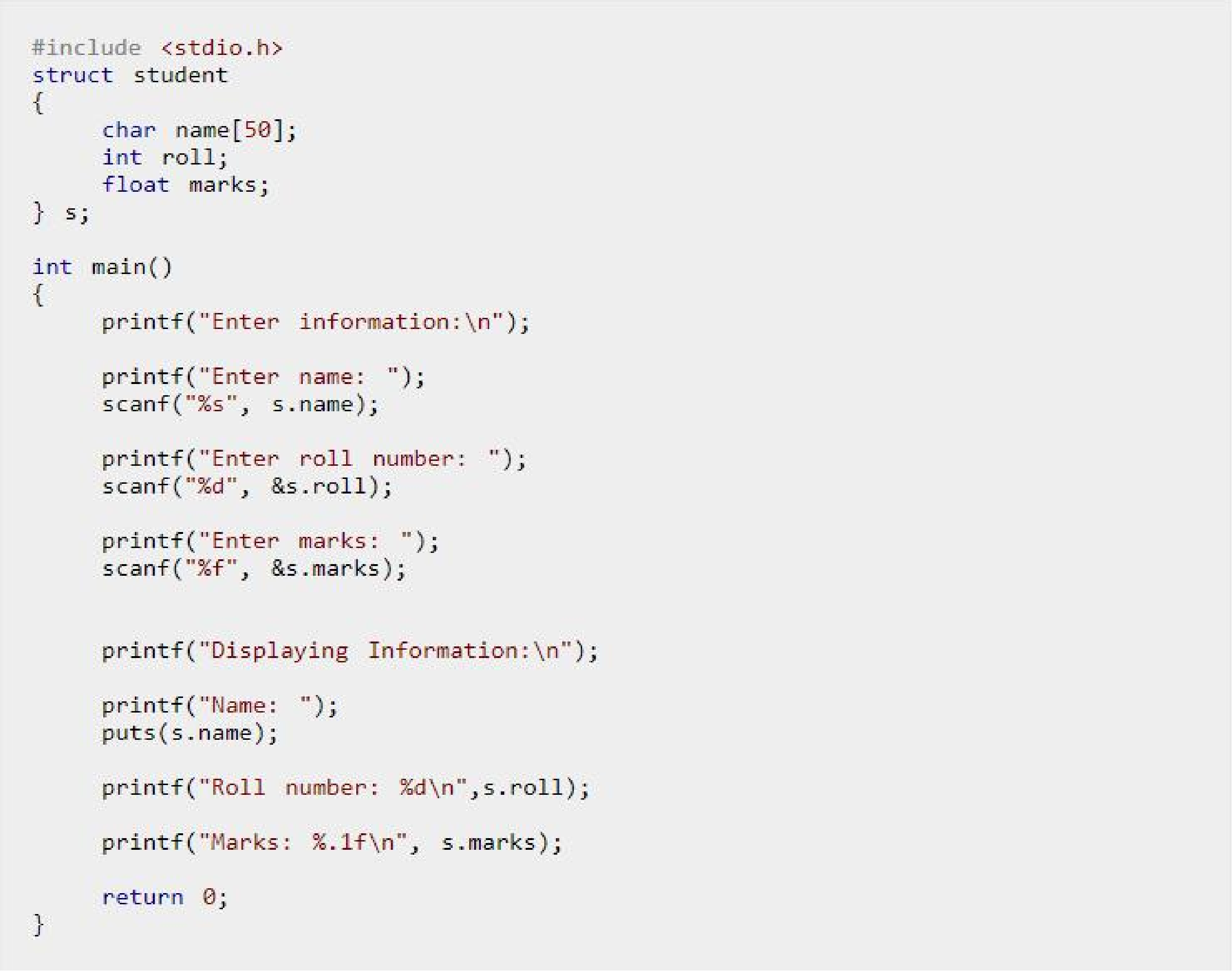


```
Enter two values:5
6

Cube value for val1 is          :125
Cube value for val2 is          :125
```

EXPERIMENT: 6 (A)

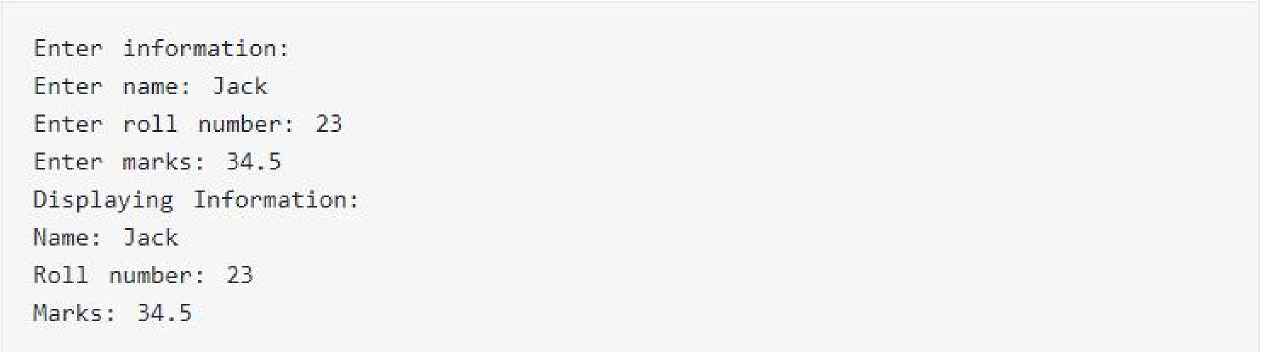
Aim: Program for implement the structure in c programming.

Code:



```
#include <stdio.h>
struct student
{
     char name[50];
     int roll;
     float marks;
} s;

int main()
{
     printf("Enter information:\n");

     printf("Enter name: ");
     scanf("%s", s.name);

     printf("Enter roll number: ");
     scanf("%d", &s.roll);

     printf("Enter marks: ");
     scanf("%f", &s.marks);


     printf("Displaying Information:\n");

     printf("Name: ");
     puts(s.name);

     printf("Roll number: %d\n",s.roll);

     printf("Marks: %.1f\n", s.marks);

     return 0;
}
```

Output:

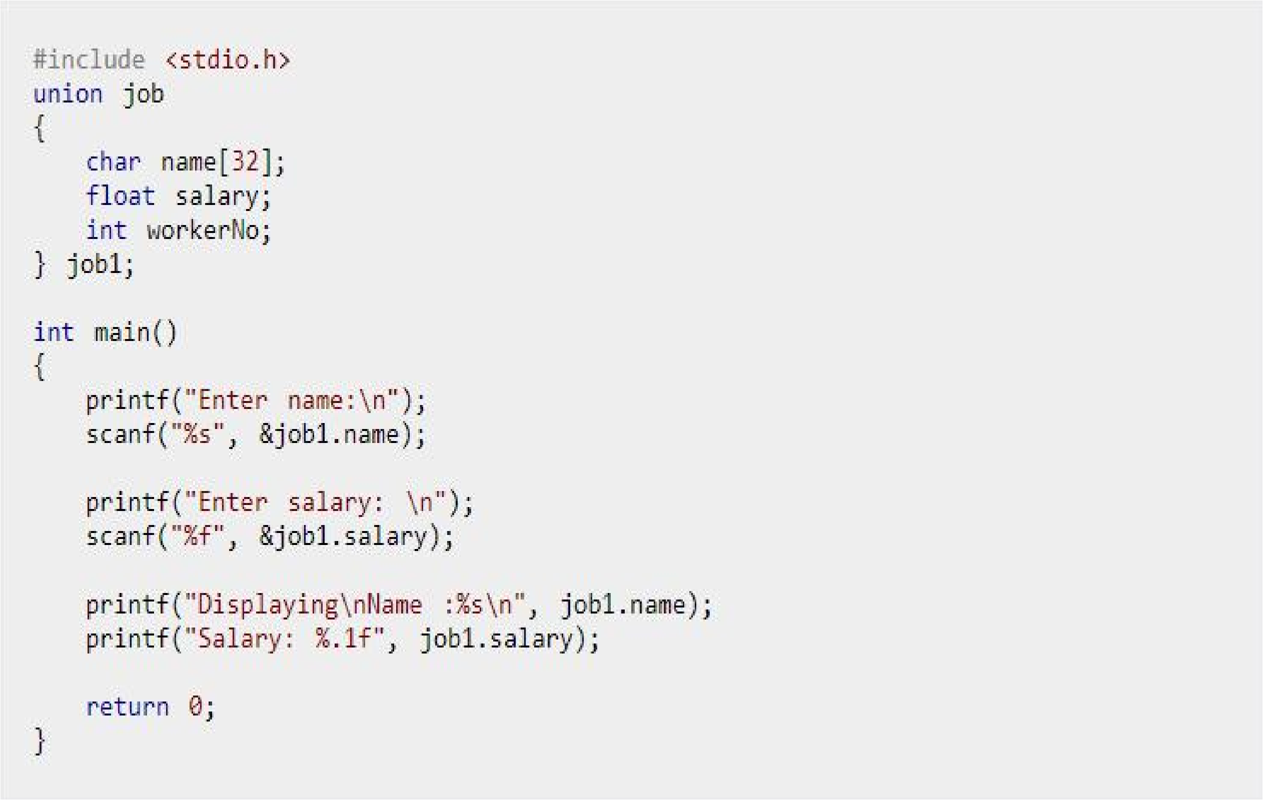


```
Enter information:
Enter name: Jack
Enter roll number: 23
Enter marks: 34.5
Displaying Information:
Name: Jack
Roll number: 23
Marks: 34.5
```

EXPERIMENT: 6 (B)

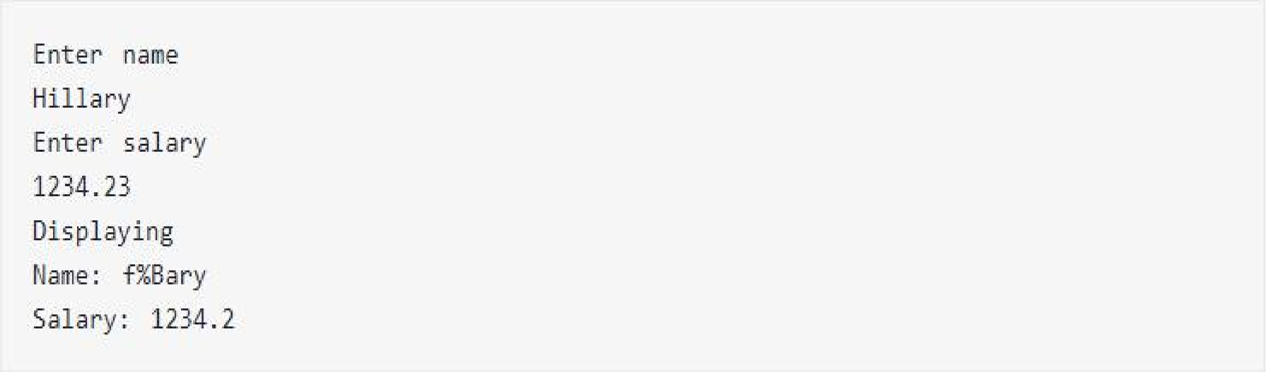
Aim: Program for implement the union in c programming.

Code:



```
#include <stdio.h>
union job
{
    char name[32];
    float salary;
    int workerNo;
} job1;

int main()
{
    printf("Enter name:\n");
    scanf("%s", &job1.name);

    printf("Enter salary: \n");
    scanf("%f", &job1.salary);

    printf("Displaying\nName :%s\n", job1.name);
    printf("Salary: %.1f", job1.salary);

    return 0;
}
```

Output:

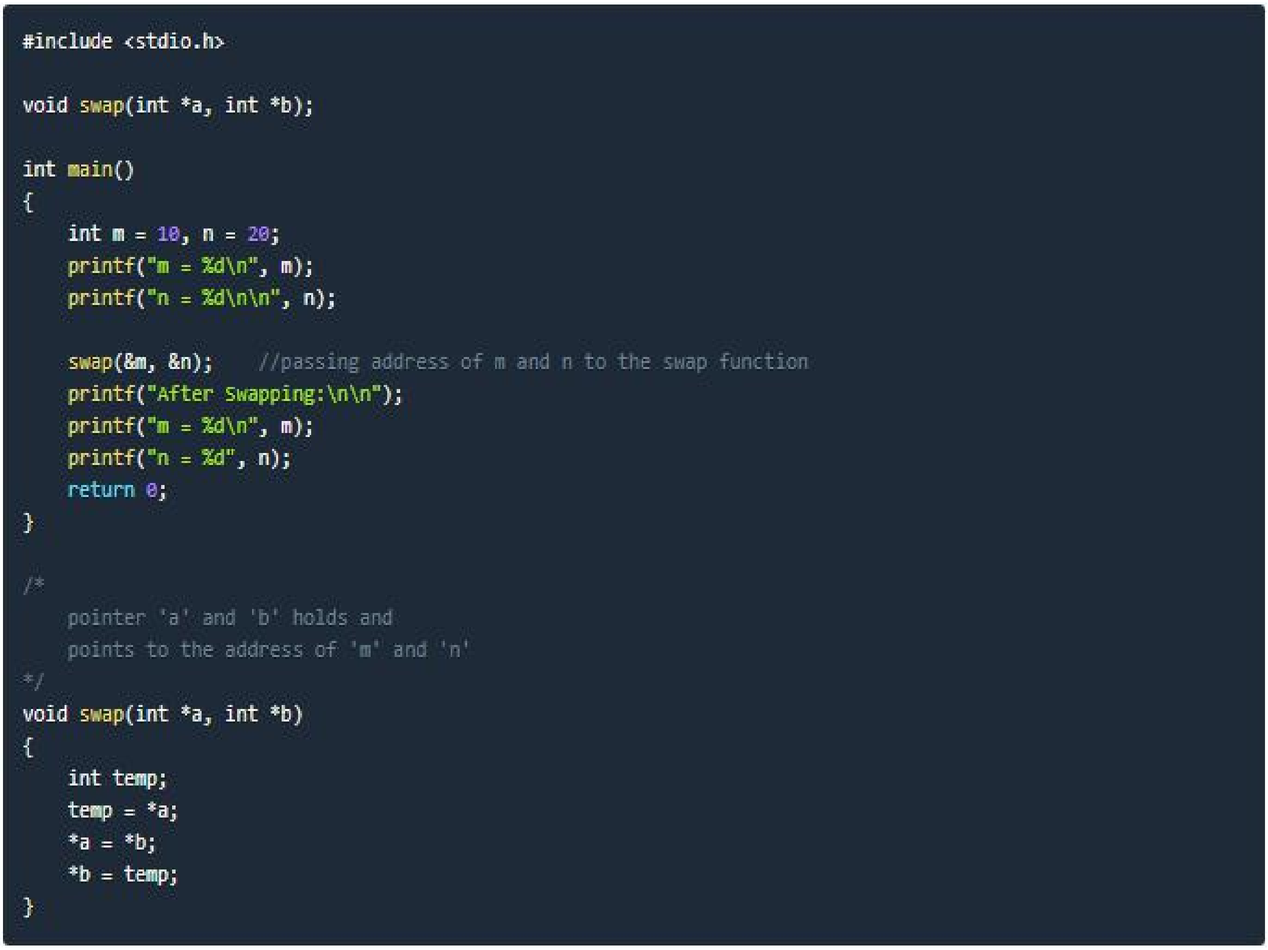


```
Enter name
Hillary
Enter salary
1234.23
Displaying
Name: f%Bary
Salary: 1234.2
```

EXPERIMENT: 7 (A)

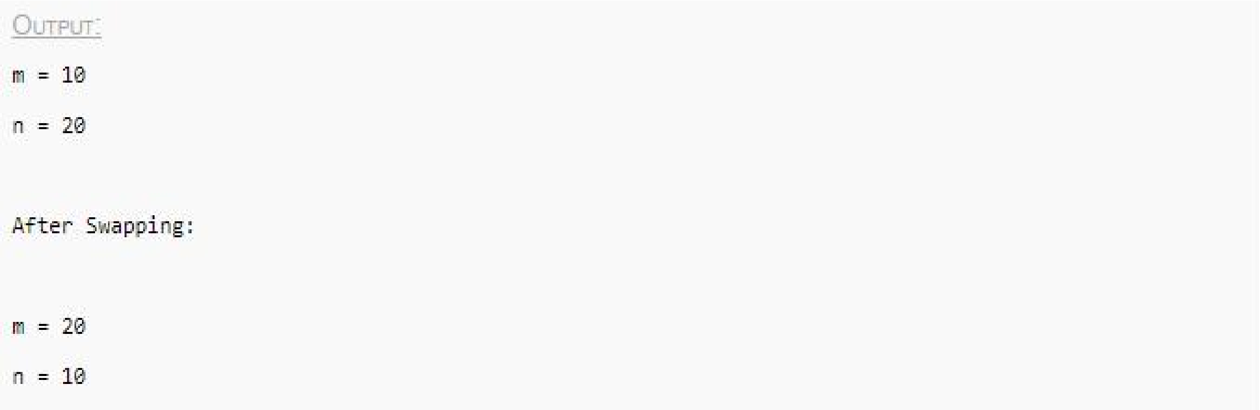
Aim: Program for implement the pointer to function in c programming.

Code:



```
#include <stdio.h>

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

int main()
{
    int m = 10, n = 20;
    printf("m = %d\n", m);
    printf("n = %d\n\n", n);

    swap(&m, &n);    //passing address of m and n to the swap function
    printf("After Swapping:\n\n");
    printf("m = %d\n", m);
    printf("n = %d", n);
    return 0;
}

/*
    pointer 'a' and 'b' holds and
    points to the address of 'm' and 'n'
*/
void swap(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
```

Output:



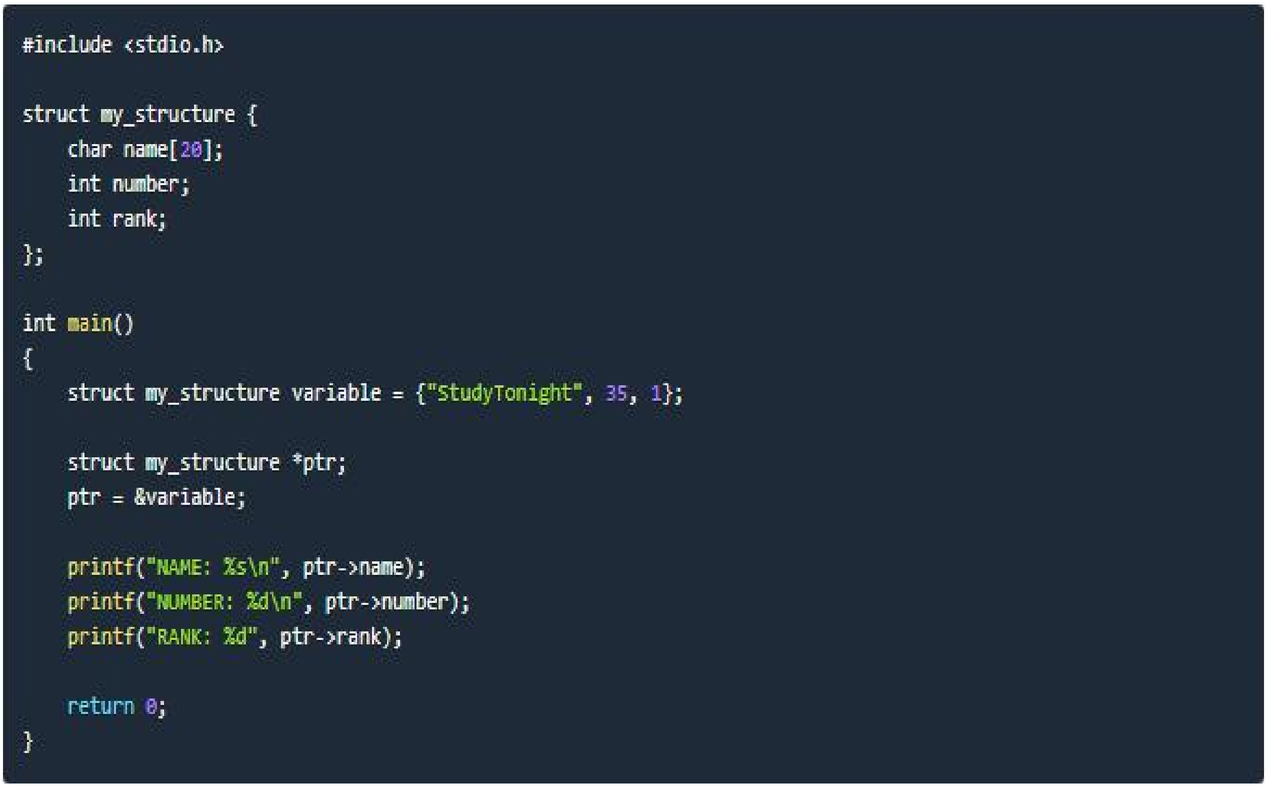
OUTPUT:

```
m = 10

n = 20


After Swapping:


m = 20

n = 10
```

EXPERIMENT: 7 (B)

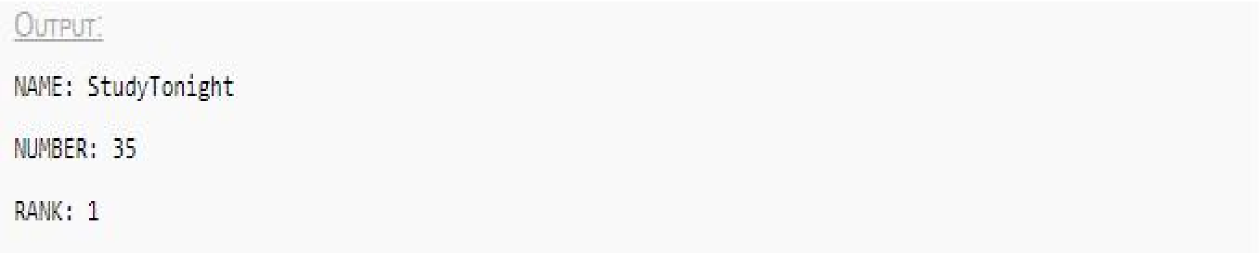
Aim: Program for implement the pointer to function in c programming.

Code:



```c
#include <stdio.h>

struct my_structure {
    char name[20];
    int number;
    int rank;
};

int main()
{
    struct my_structure variable = {"StudyTonight", 35, 1};

    struct my_structure *ptr;
    ptr = &variable;

    printf("NAME: %s\n", ptr->name);
    printf("NUMBER: %d\n", ptr->number);
    printf("RANK: %d", ptr->rank);

    return 0;
}
```

Output:

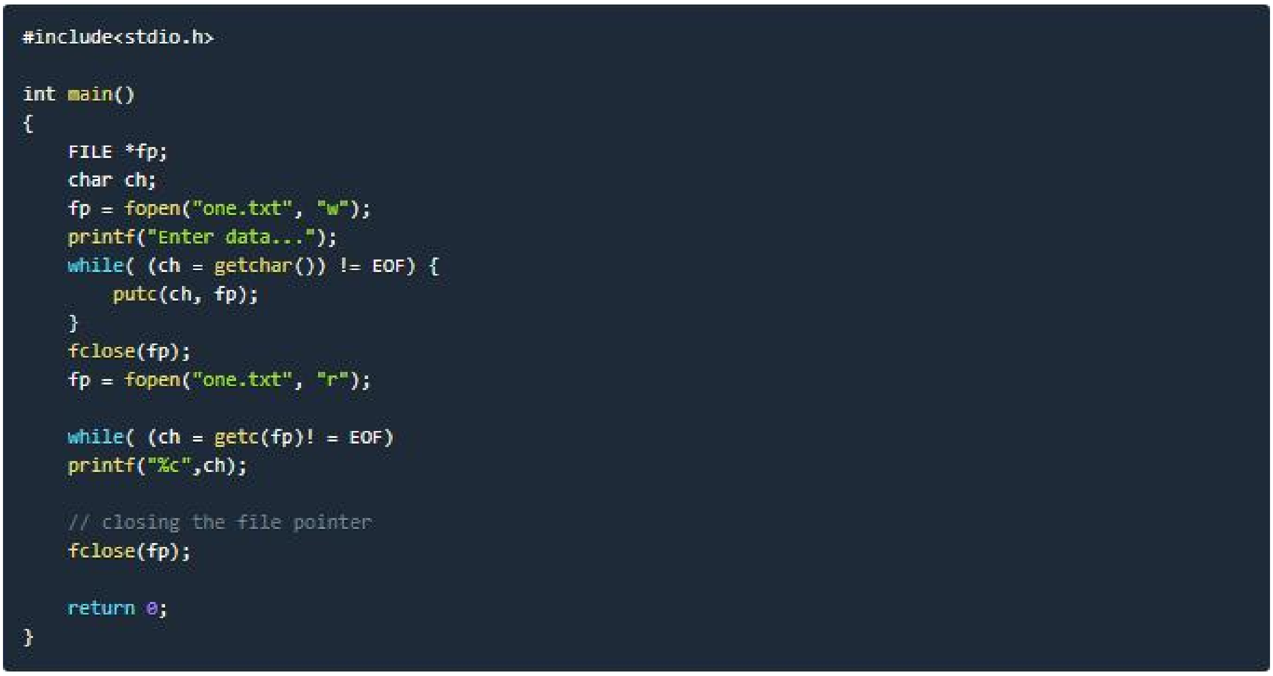


```
OUTPUT:

NAME: StudyTonight

NUMBER: 35

RANK: 1
```

EXPERIMENT: 8 (A)

Aim: Program for implement input/output operation on a file.

Code:

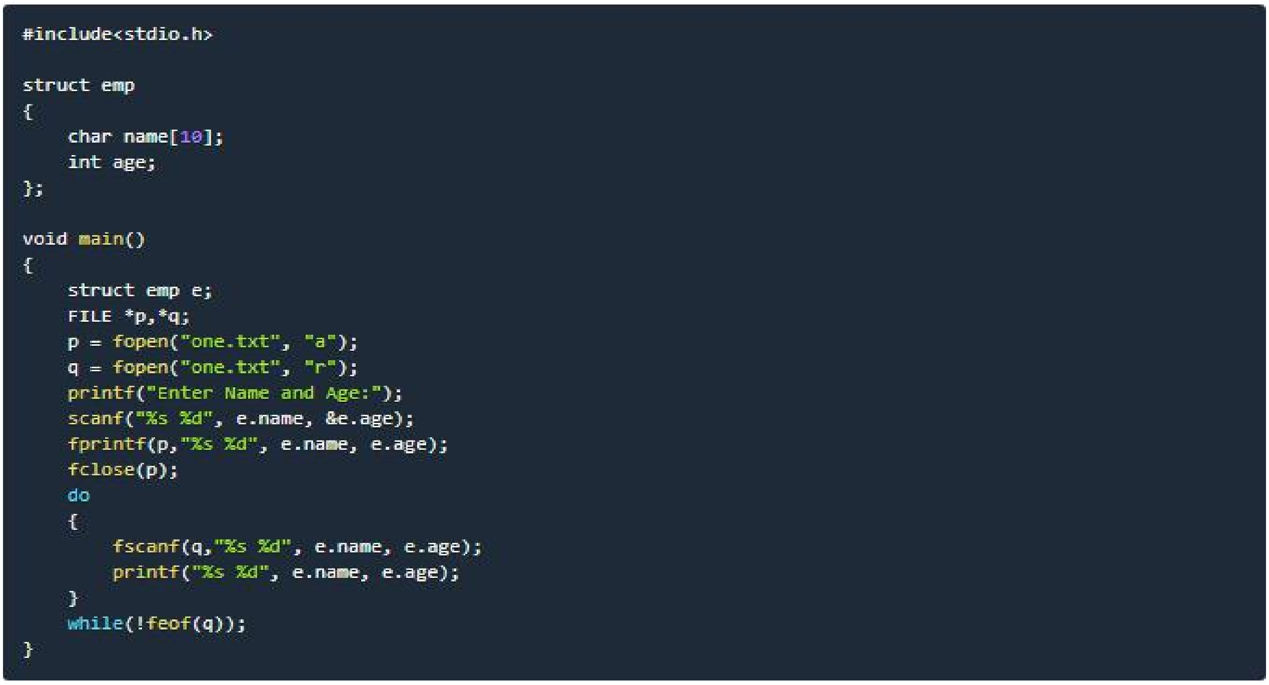


```
#include<stdio.h>

int main()
{
    FILE *fp;
    char ch;
    fp = fopen("one.txt", "w");
    printf("Enter data...");
    while( (ch = getchar()) != EOF) {
        putc(ch, fp);
    }
    fclose(fp);
    fp = fopen("one.txt", "r");

    while( (ch = getc(fp)! = EOF)
    printf("%c",ch);

    // closing the file pointer
    fclose(fp);

    return 0;
}
```

EXPERIMENT: 8 (B)

Aim: Program for implement reading and writing on a file using fprintf and fscanf.

Code:

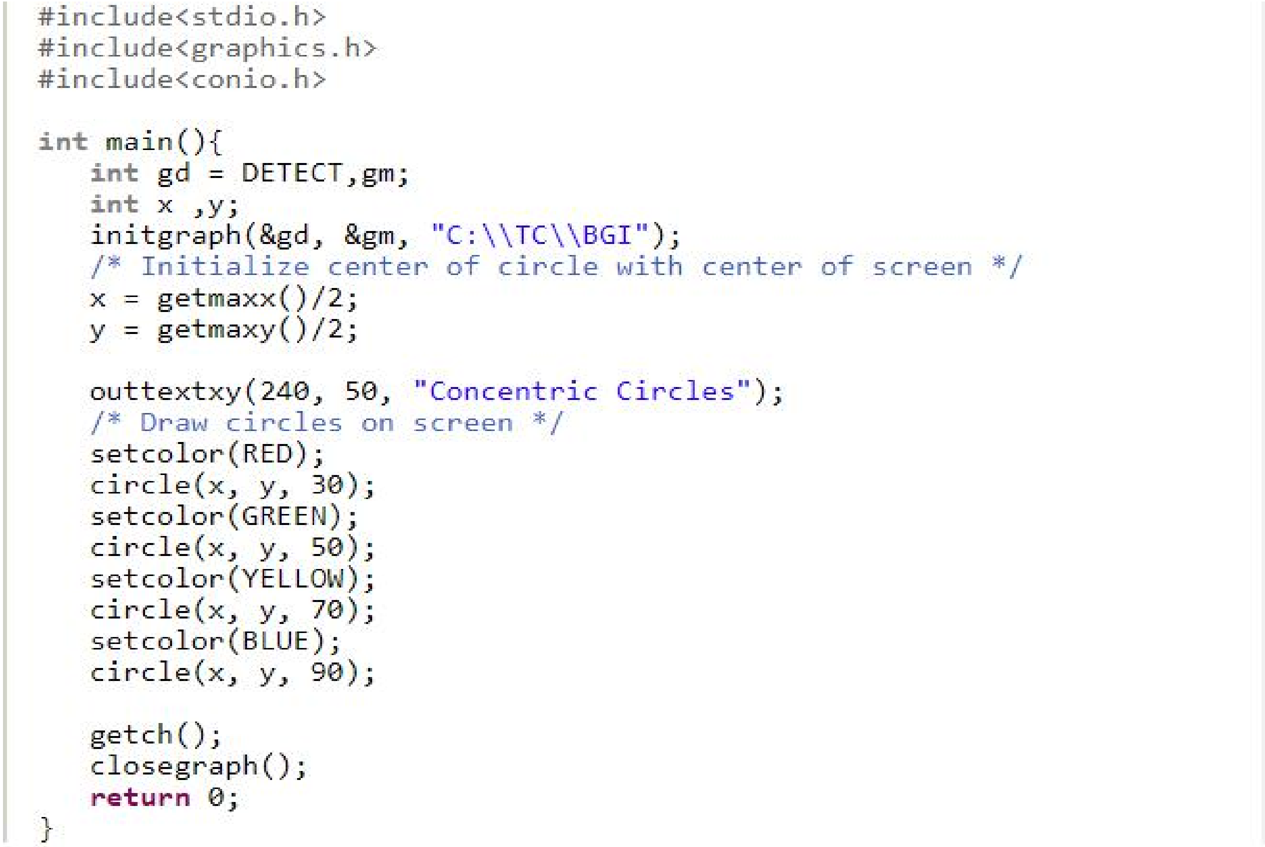


```
#include<stdio.h>

struct emp
{
    char name[10];
    int age;
};

void main()
{
    struct emp e;
    FILE *p,*q;
    p = fopen("one.txt", "a");
    q = fopen("one.txt", "r");
    printf("Enter Name and Age:");
    scanf("%s %d", e.name, &e.age);
    fprintf(p,"%s %d", e.name, e.age);
    fclose(p);
    do
    {
        fscanf(q,"%s %d", e.name, e.age);
        printf("%s %d", e.name, e.age);
    }
    while(!feof(q));
}
```

EXPERIMENT: 9 (A)

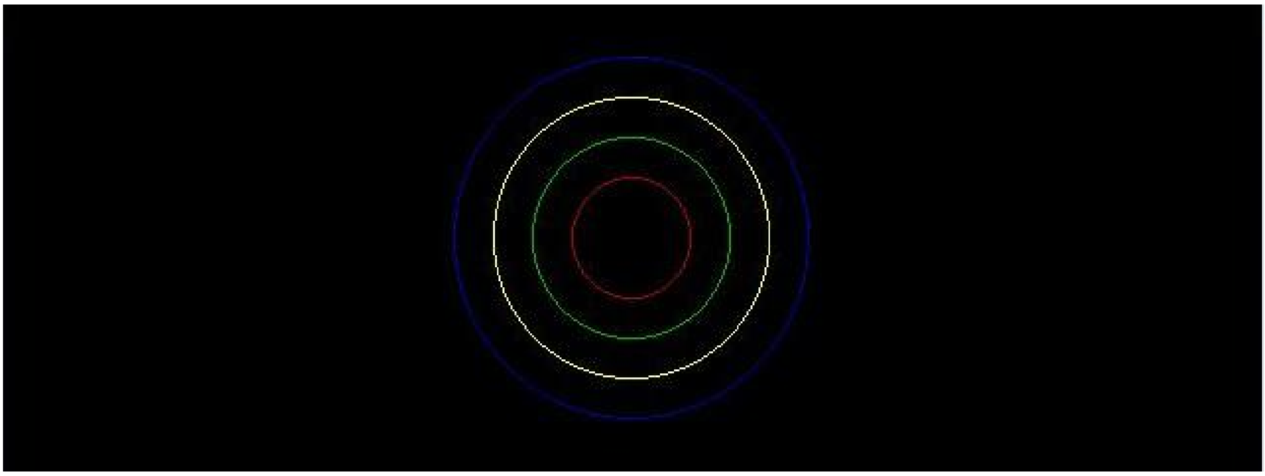
Aim: Program to draw concentric circle on the screen by using graphics.

Code:



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

int main(){
   int gd = DETECT,gm;
   int x ,y;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
   /* Initialize center of circle with center of screen */
   x = getmaxx()/2;
   y = getmaxy()/2;

   outtextxy(240, 50, "Concentric Circles");
   /* Draw circles on screen */
   setcolor(RED);
   circle(x, y, 30);
   setcolor(GREEN);
   circle(x, y, 50);
   setcolor(YELLOW);
   circle(x, y, 70);
   setcolor(BLUE);
   circle(x, y, 90);

   getch();
   closegraph();
   return 0;
}
```

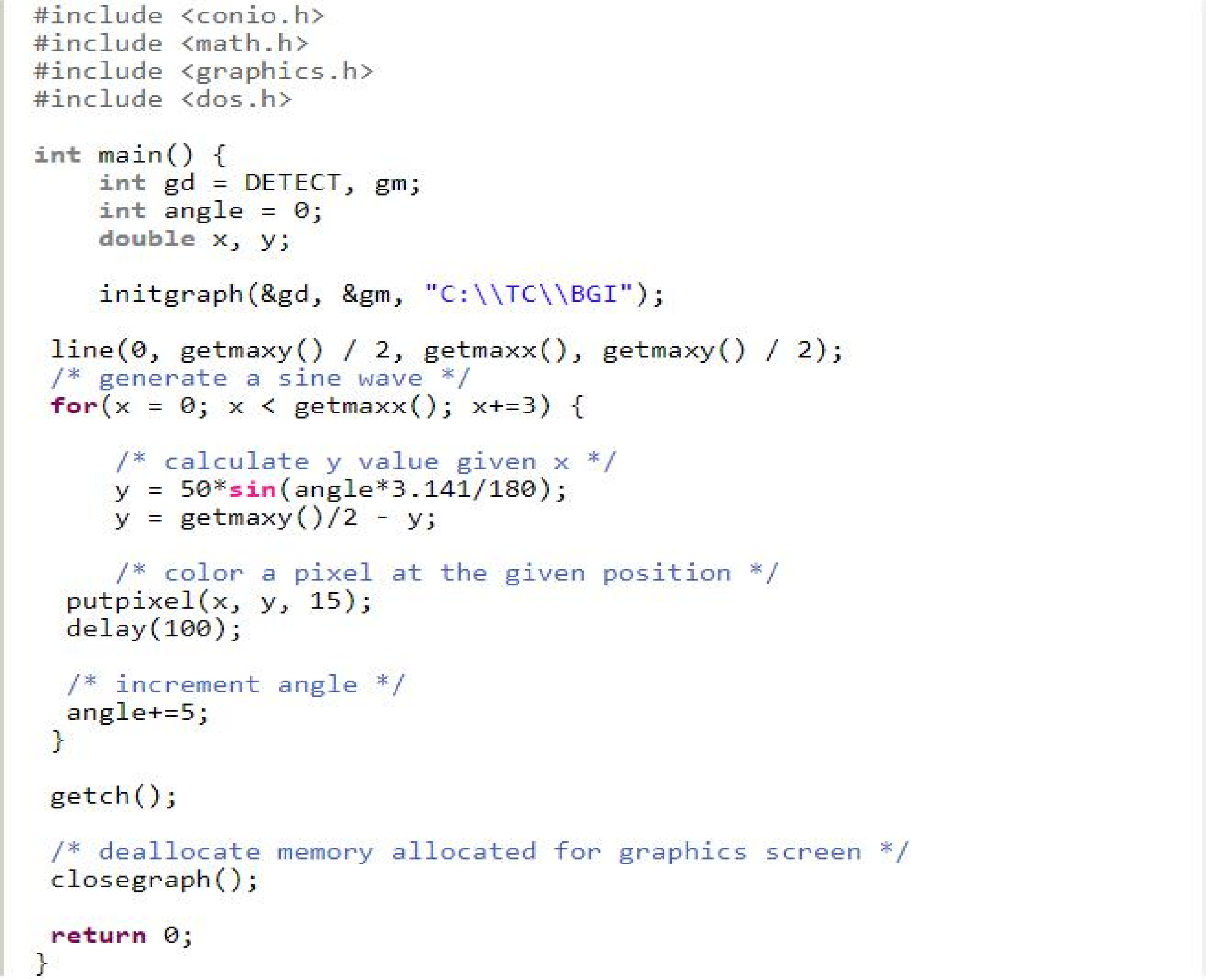
Output:



EXPERIMENT: 9 (B)

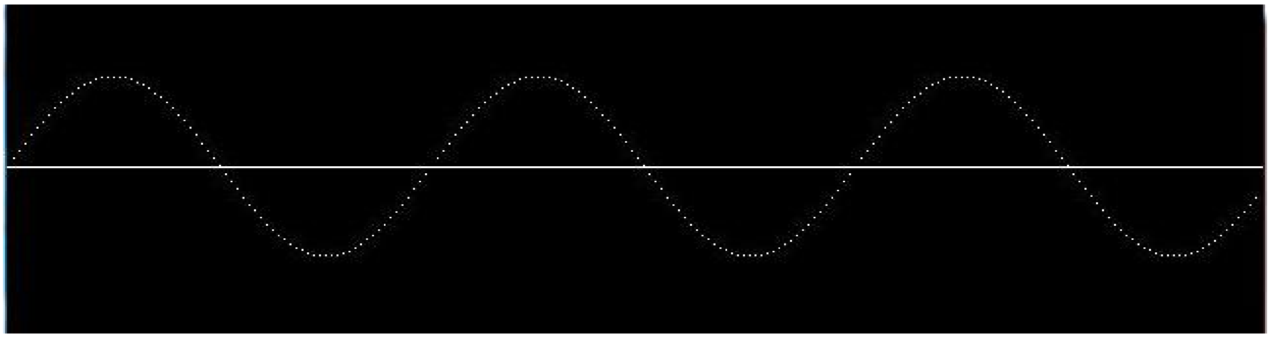
Aim: Program to draw sine wave by using graphics.

Code:



```
#include <conio.h>
#include <math.h>
#include <graphics.h>
#include <dos.h>

int main() {
    int gd = DETECT, gm;
    int angle = 0;
    double x, y;

    initgraph(&gd, &gm, "C:\\TC\\BGI");

 line(0, getmaxy() / 2, getmaxx(), getmaxy() / 2);
 /* generate a sine wave */
 for(x = 0; x < getmaxx(); x+=3) {

     /* calculate y value given x */
     y = 50*sin(angle*3.141/180);
     y = getmaxy()/2 - y;

     /* color a pixel at the given position */
  putpixel(x, y, 15);
  delay(100);

  /* increment angle */
  angle+=5;
 }

 getch();

 /* deallocate memory allocated for graphics screen */
 closegraph();

 return 0;
}
```

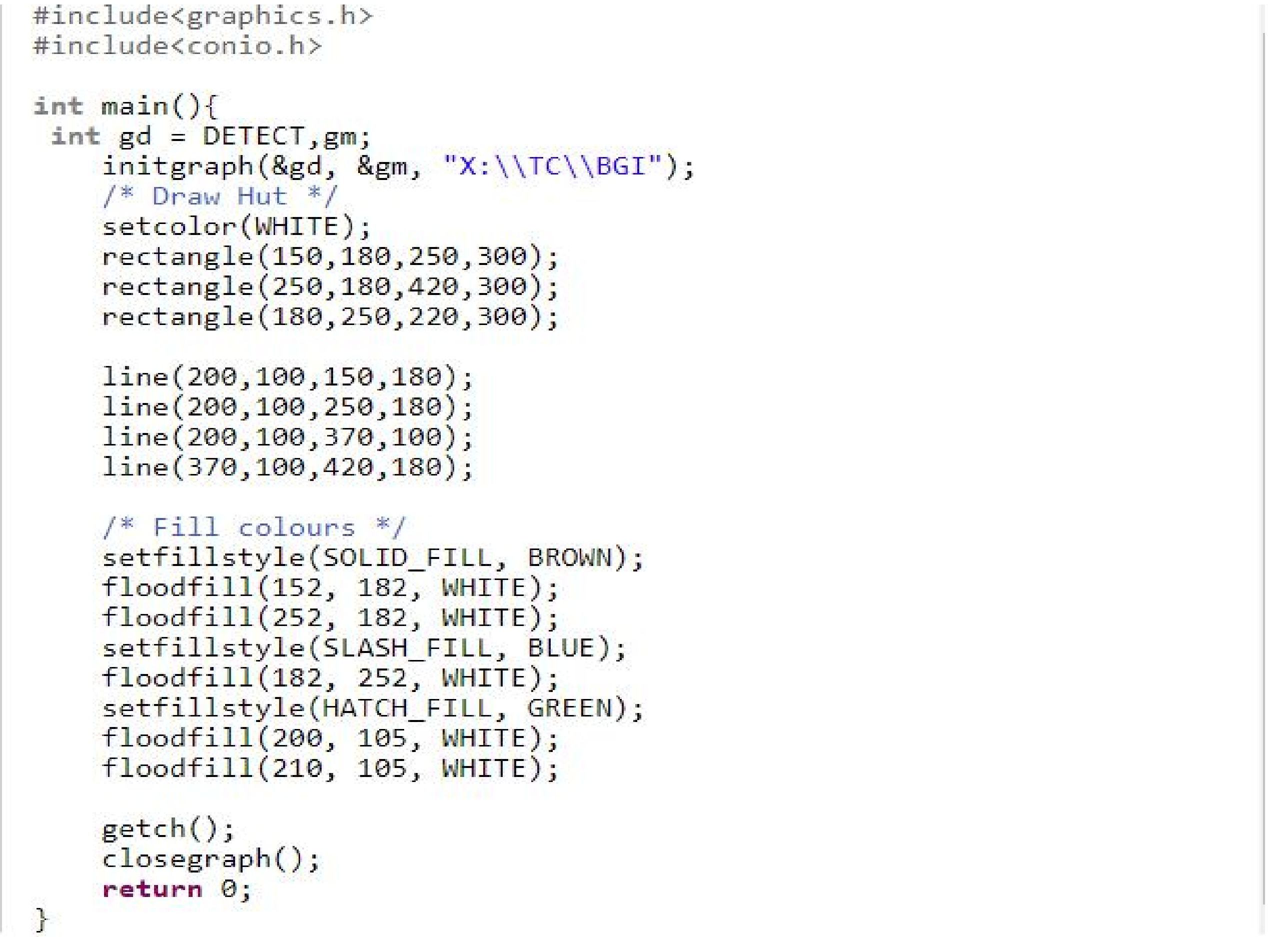
Output:



EXPERIMENT: 9 (C)

Aim: Program to draw a hut and color it by using graphics.

Code:



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

int main(){
 int gd = DETECT,gm;
    initgraph(&gd, &gm, "X:\\TC\\BGI");
    /* Draw Hut */
    setcolor(WHITE);
    rectangle(150,180,250,300);
    rectangle(250,180,420,300);
    rectangle(180,250,220,300);

    line(200,100,150,180);
    line(200,100,250,180);
    line(200,100,370,100);
    line(370,100,420,180);

    /* Fill colours */
    setfillstyle(SOLID_FILL, BROWN);
    floodfill(152, 182, WHITE);
    floodfill(252, 182, WHITE);
    setfillstyle(SLASH_FILL, BLUE);
    floodfill(182, 252, WHITE);
    setfillstyle(HATCH_FILL, GREEN);
    floodfill(200, 105, WHITE);
    floodfill(210, 105, WHITE);

    getch();
    closegraph();
    return 0;
}
```

Output: