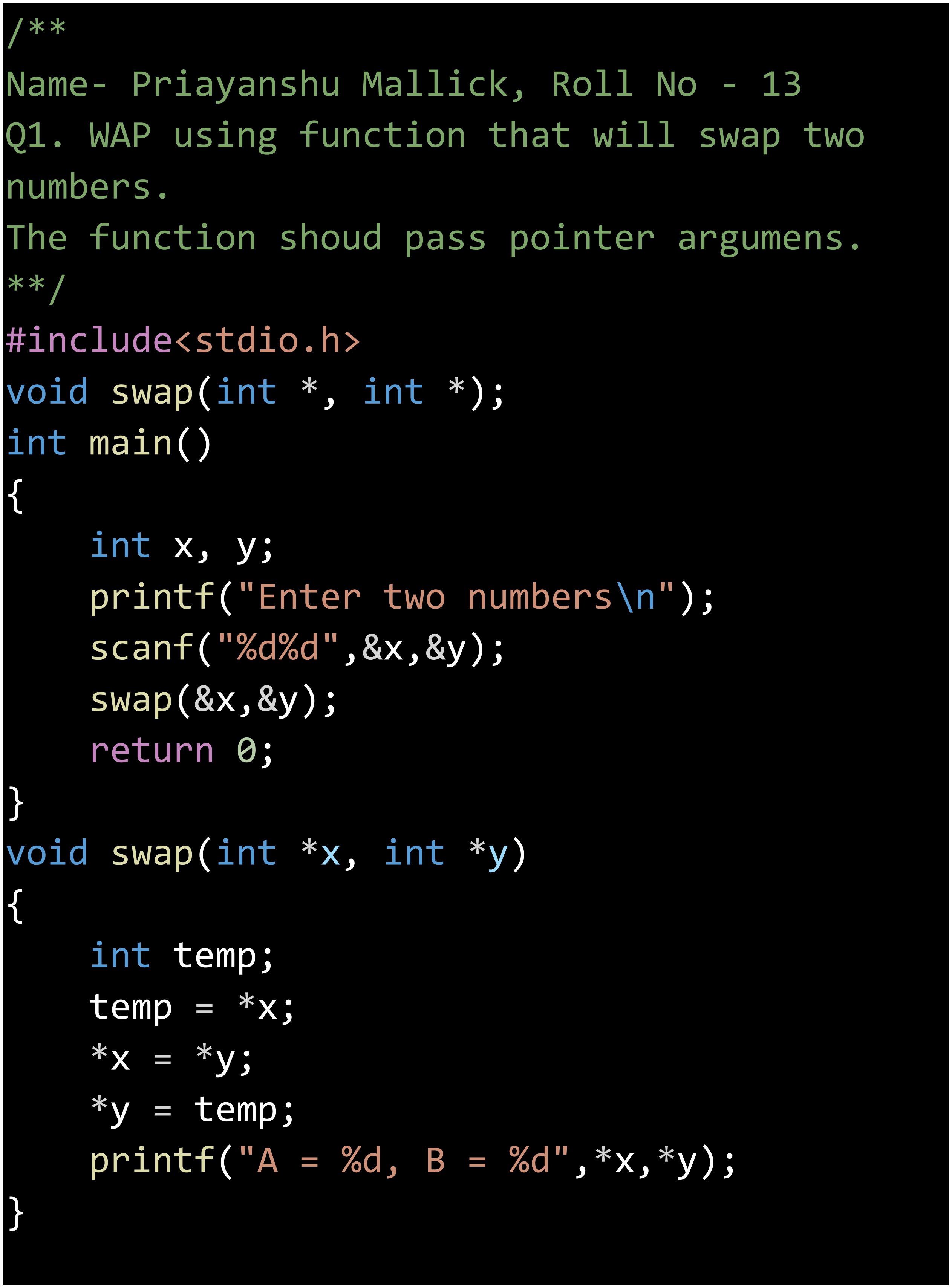
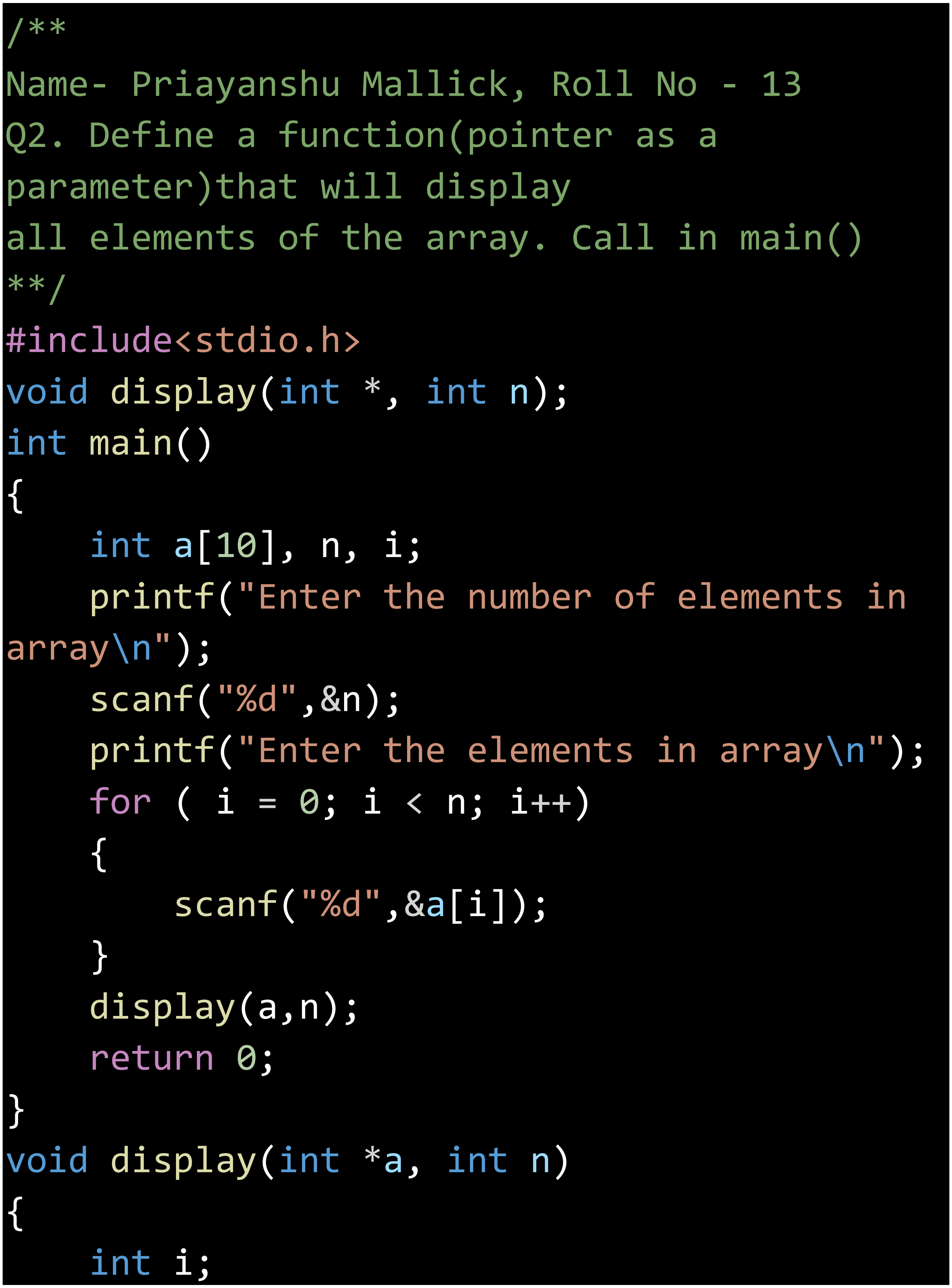
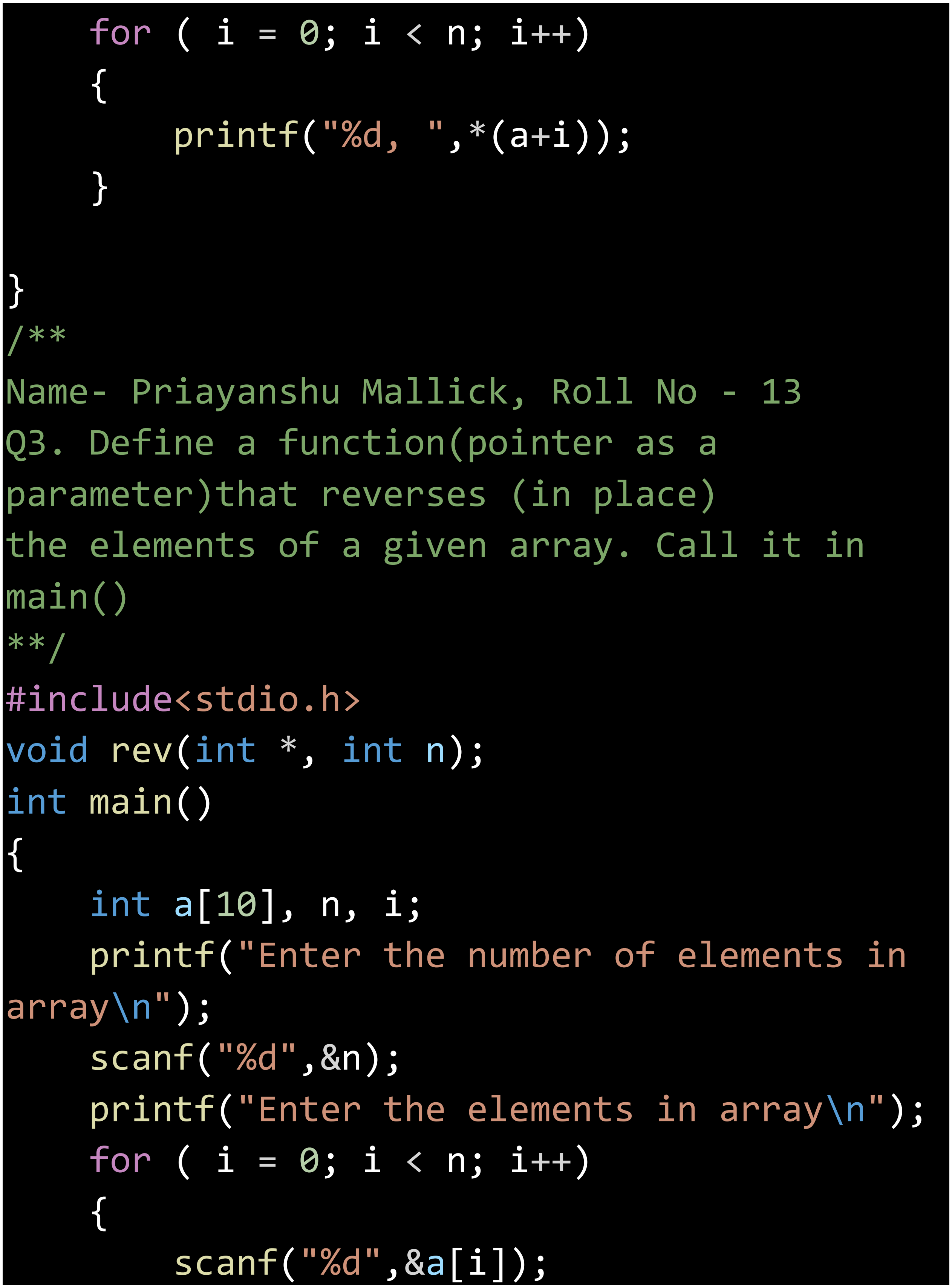


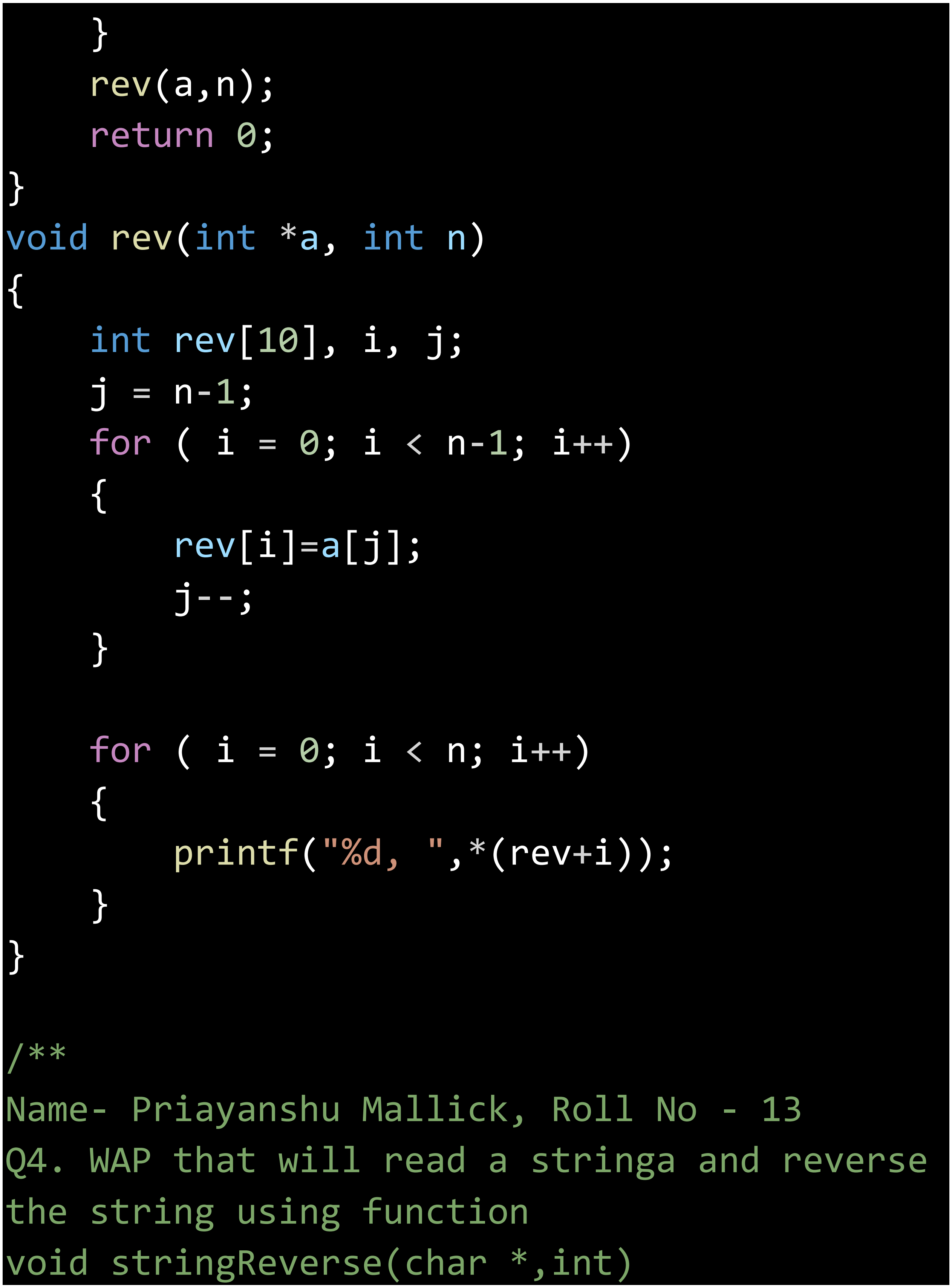
```
/**
Name- Priayanshu Mallick, Roll No - 13
Q1. WAP using function that will swap two
numbers.
The function shoud pass pointer argumens.
**/
#include<stdio.h>
void swap(int *, int *);
int main()
{
    int x, y;
    printf("Enter two numbers\n");
    scanf("%d%d",&x,&y);
    swap(&x,&y);
    return 0;
}
void swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
    printf("A = %d, B = %d",*x,*y);
}
```



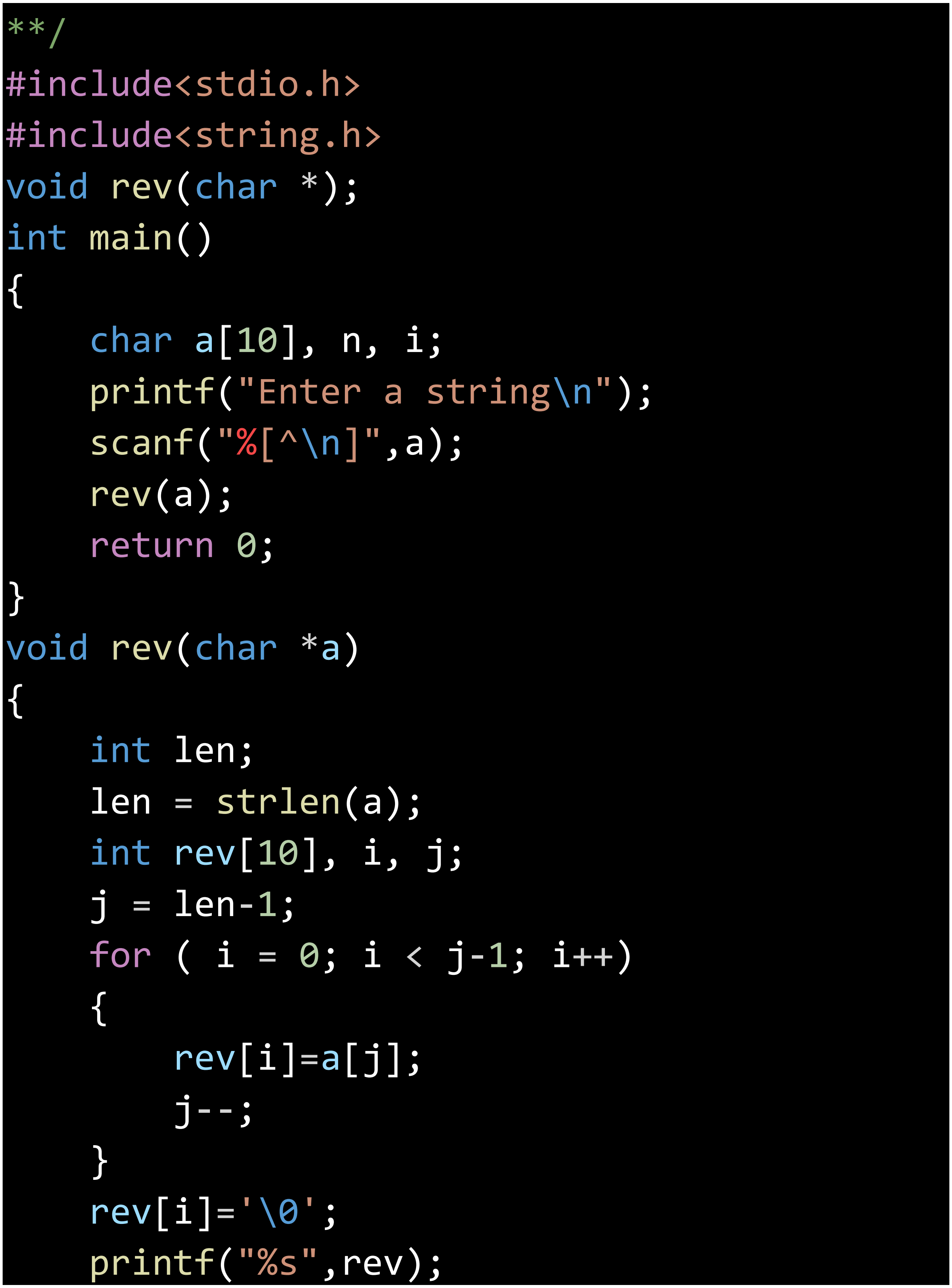
```c
/**
Name- Priayanshu Mallick, Roll No - 13
Q2. Define a function(pointer as a
parameter)that will display
all elements of the array. Call in main()
**/
#include<stdio.h>
void display(int *, int n);
int main()
{
    int a[10], n, i;
    printf("Enter the number of elements in
array\n");
    scanf("%d",&n);
    printf("Enter the elements in array\n");
    for ( i = 0; i < n; i++)
    {
        scanf("%d",&a[i]);
    }
    display(a,n);
    return 0;
}
void display(int *a, int n)
{
    int i;
```



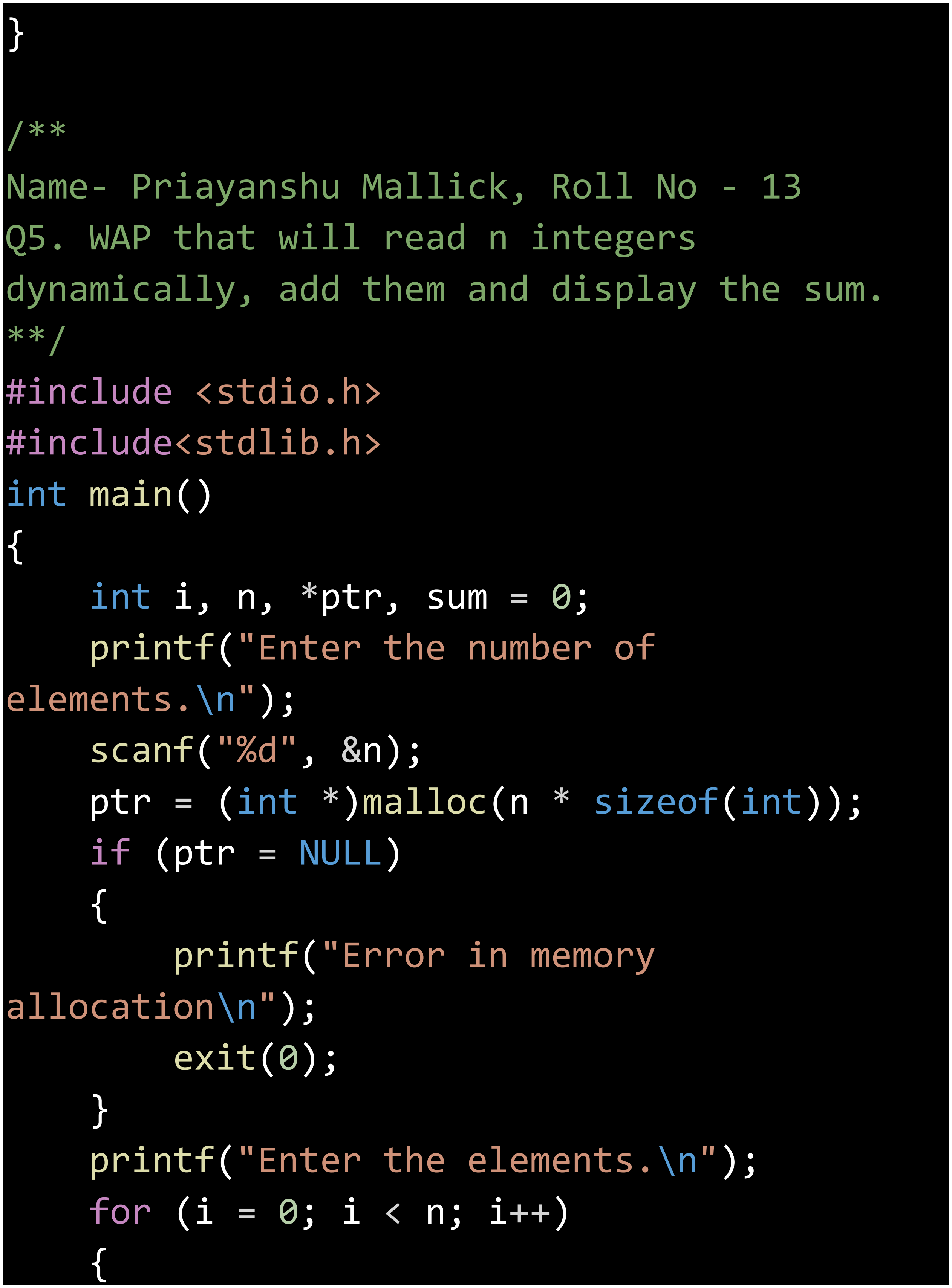
```
    for ( i = 0; i < n; i++)
    {
        printf("%d, ",*(a+i));
    }

}
/**
Name- Priayanshu Mallick, Roll No - 13
Q3. Define a function(pointer as a
parameter)that reverses (in place)
the elements of a given array. Call it in
main()
**/
#include<stdio.h>
void rev(int *, int n);
int main()
{
    int a[10], n, i;
    printf("Enter the number of elements in array\n");
    scanf("%d",&n);
    printf("Enter the elements in array\n");
    for ( i = 0; i < n; i++)
    {
        scanf("%d",&a[i]);
```



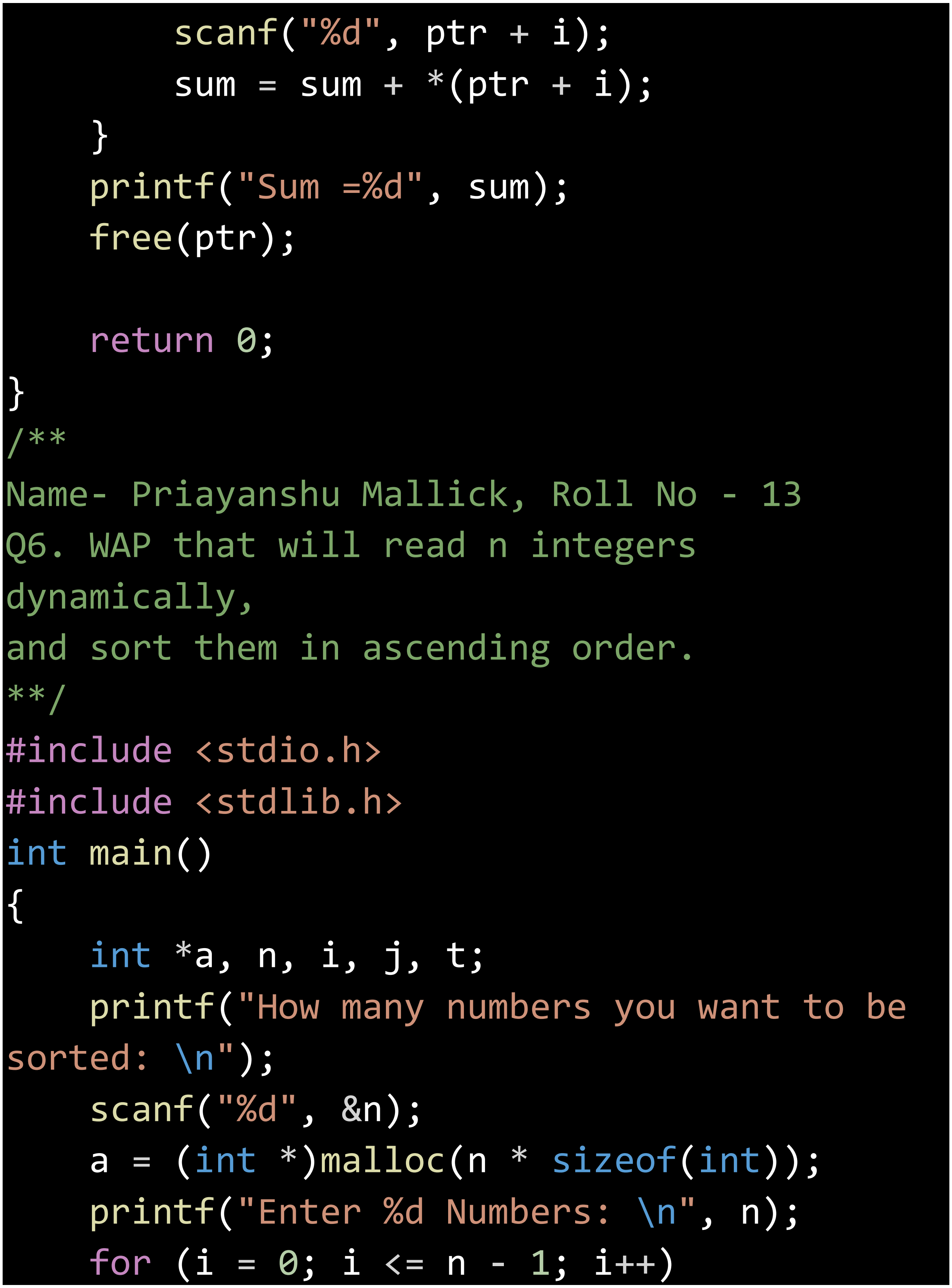
```c
    }
    rev(a,n);
    return 0;
}
void rev(int *a, int n)
{
    int rev[10], i, j;
    j = n-1;
    for ( i = 0; i < n-1; i++)
    {
        rev[i]=a[j];
        j--;
    }

    for ( i = 0; i < n; i++)
    {
        printf("%d, ",*(rev+i));
    }
}

/**
Name- Priayanshu Mallick, Roll No - 13
Q4. WAP that will read a stringa and reverse
the string using function
void stringReverse(char *,int)
```



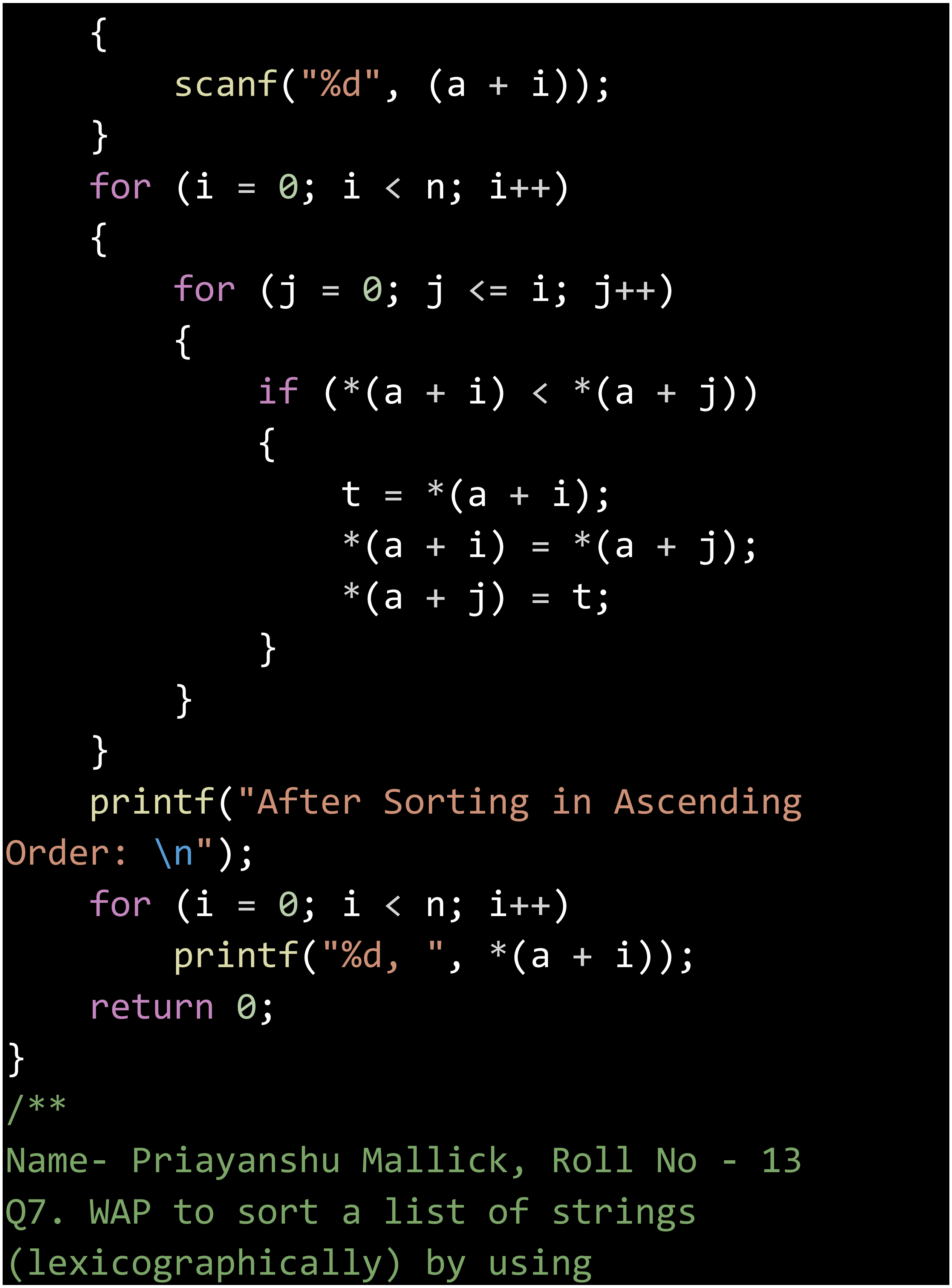
```
**/
#include<stdio.h>
#include<string.h>
void rev(char *);
int main()
{
    char a[10], n, i;
    printf("Enter a string\n");
    scanf("%[^\n]",a);
    rev(a);
    return 0;
}
void rev(char *a)
{
    int len;
    len = strlen(a);
    int rev[10], i, j;
    j = len-1;
    for ( i = 0; i < j-1; i++)
    {
        rev[i]=a[j];
        j--;
    }
    rev[i]='\0';
    printf("%s",rev);
```



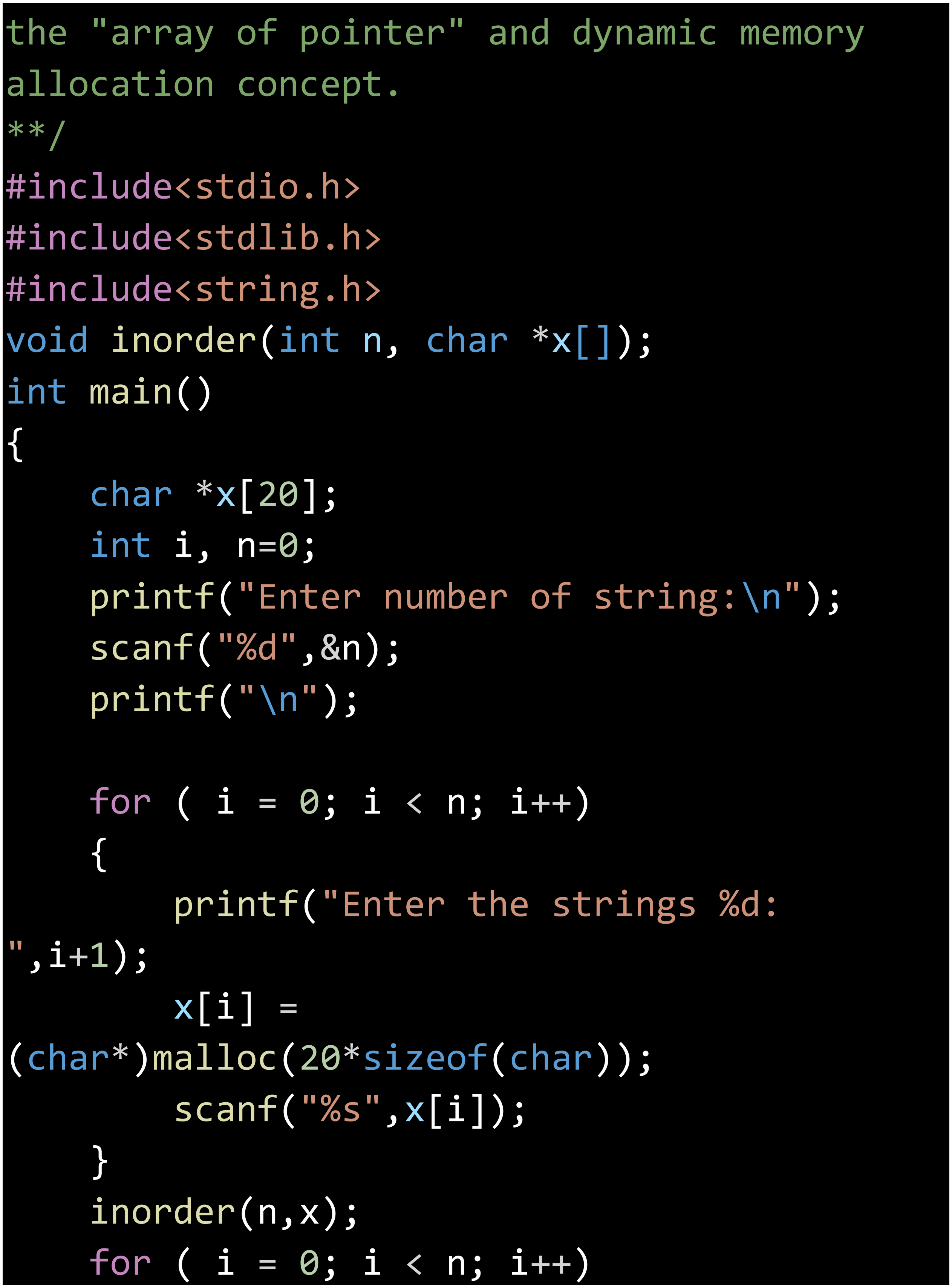
```
}

/**
Name- Priayanshu Mallick, Roll No - 13
Q5. WAP that will read n integers
dynamically, add them and display the sum.
**/
#include <stdio.h>
#include<stdlib.h>
int main()
{
    int i, n, *ptr, sum = 0;
    printf("Enter the number of
elements.\n");
    scanf("%d", &n);
    ptr = (int *)malloc(n * sizeof(int));
    if (ptr = NULL)
    {
        printf("Error in memory
allocation\n");
        exit(0);
    }
    printf("Enter the elements.\n");
    for (i = 0; i < n; i++)
    {
```



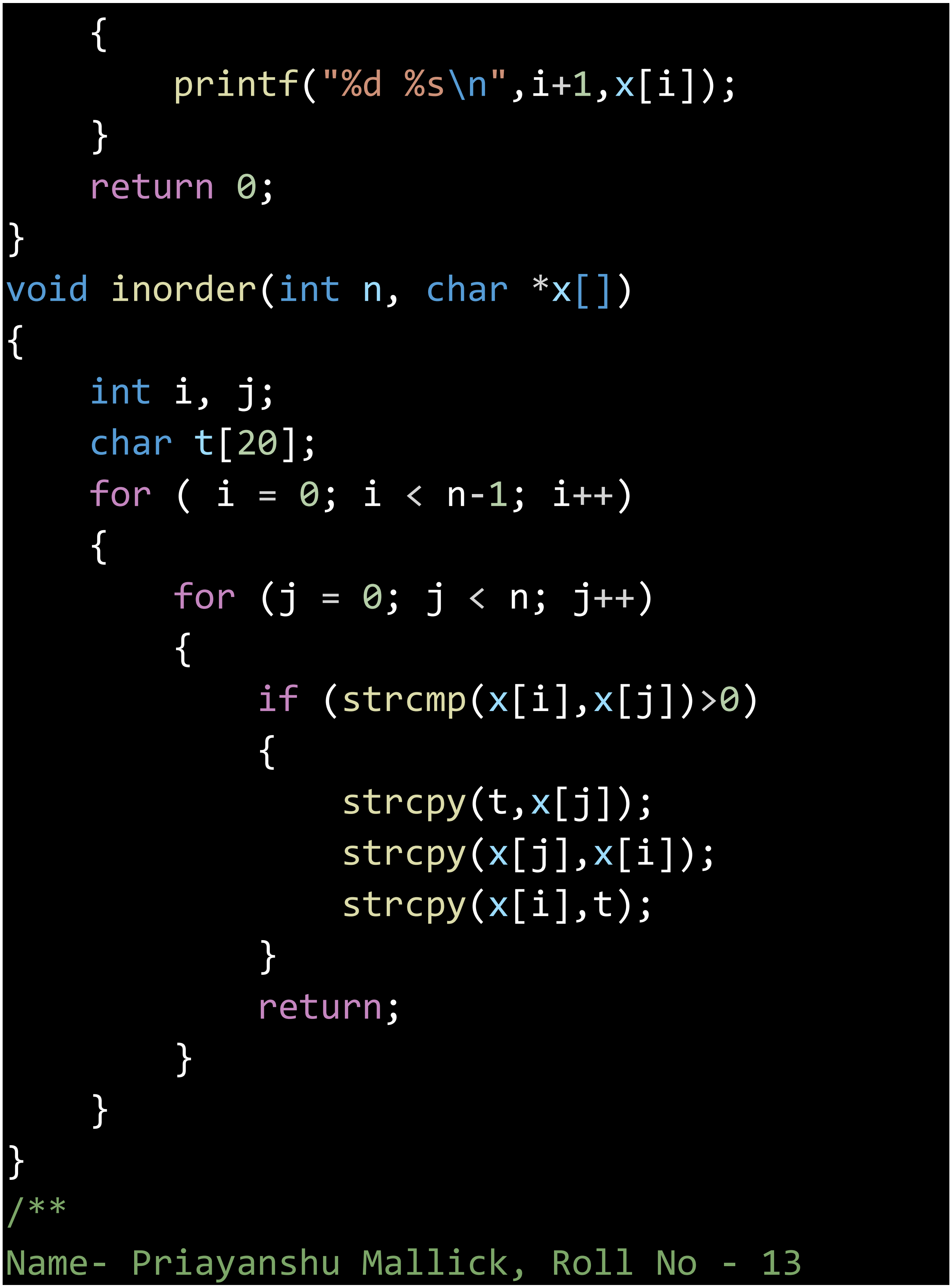
```
        scanf("%d", ptr + i);
        sum = sum + *(ptr + i);
    }
    printf("Sum =%d", sum);
    free(ptr);

    return 0;
}
/**
Name- Priayanshu Mallick, Roll No - 13
Q6. WAP that will read n integers
dynamically,
and sort them in ascending order.
**/
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int *a, n, i, j, t;
    printf("How many numbers you want to be
sorted: \n");
    scanf("%d", &n);
    a = (int *)malloc(n * sizeof(int));
    printf("Enter %d Numbers: \n", n);
    for (i = 0; i <= n - 1; i++)
```



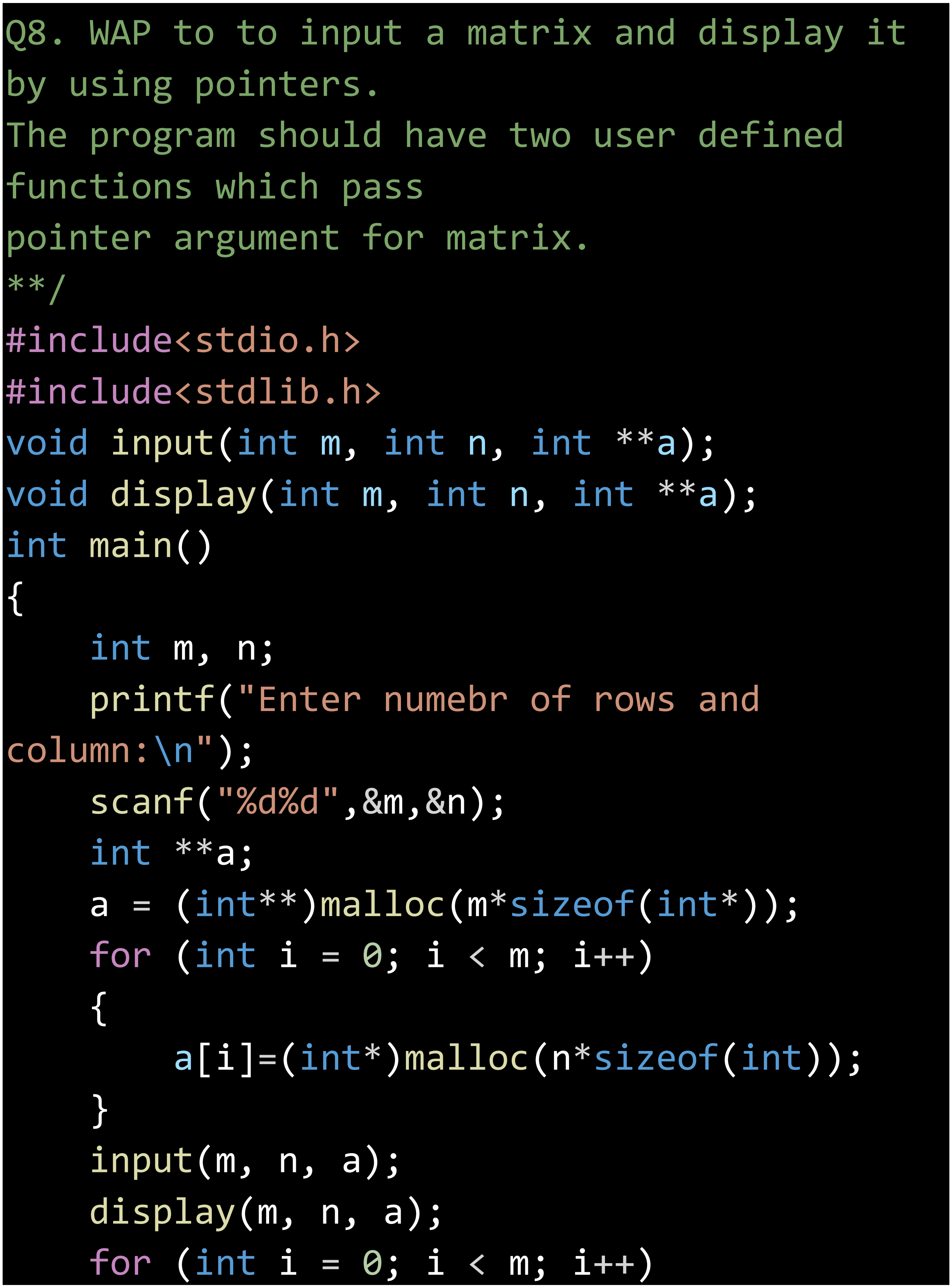
```
    {
        scanf("%d", (a + i));
    }
    for (i = 0; i < n; i++)
    {
        for (j = 0; j <= i; j++)
        {
            if (*(a + i) < *(a + j))
            {
                t = *(a + i);
                *(a + i) = *(a + j);
                *(a + j) = t;
            }
        }
    }
    printf("After Sorting in Ascending
Order: \n");
    for (i = 0; i < n; i++)
        printf("%d, ", *(a + i));
    return 0;
}
/**
Name- Priayanshu Mallick, Roll No - 13
Q7. WAP to sort a list of strings
(lexicographically) by using
```



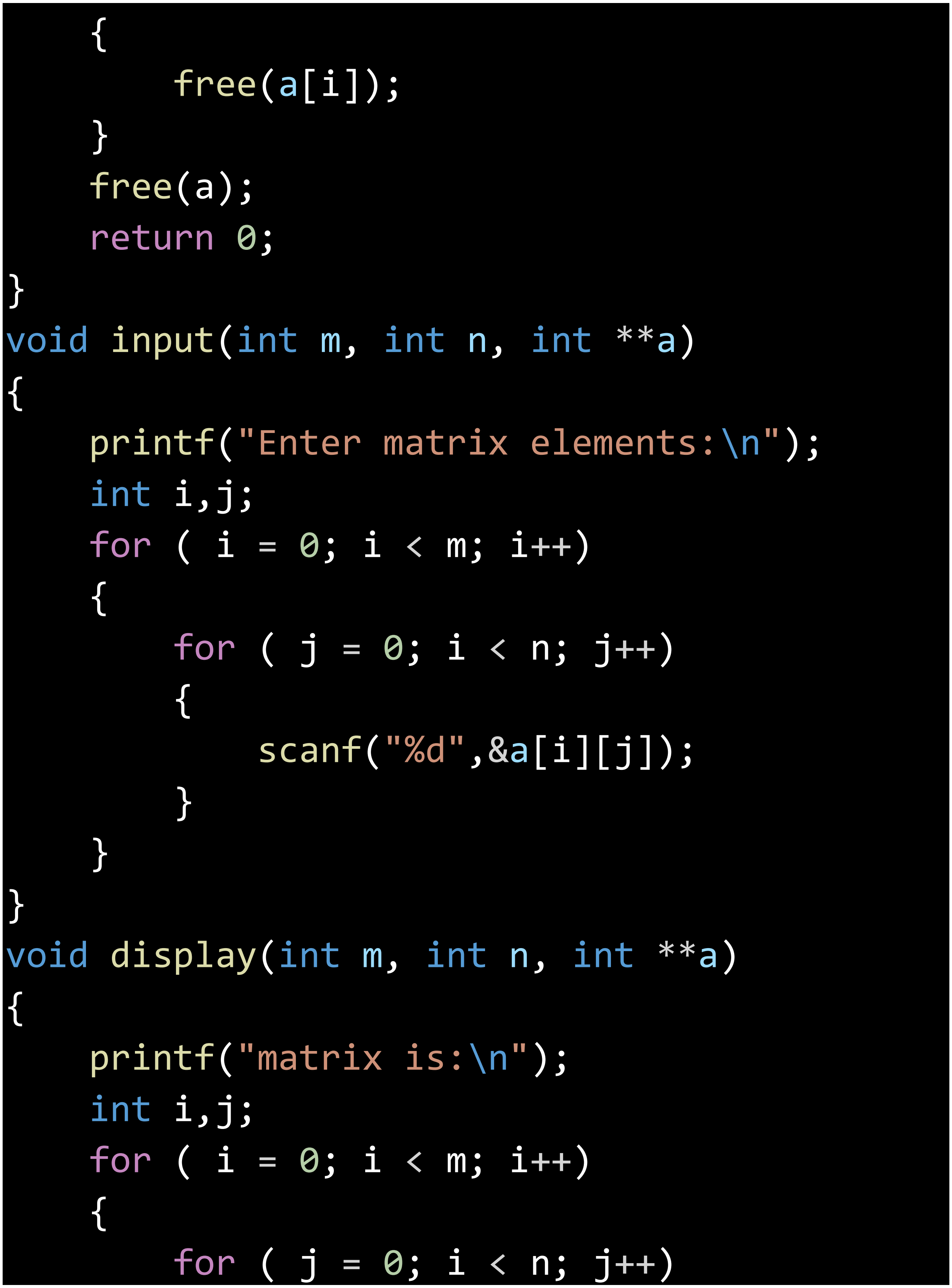
```
the "array of pointer" and dynamic memory
allocation concept.
**/
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void inorder(int n, char *x[]);
int main()
{
    char *x[20];
    int i, n=0;
    printf("Enter number of string:\n");
    scanf("%d",&n);
    printf("\n");

    for ( i = 0; i < n; i++)
    {
        printf("Enter the strings %d:
",i+1);
        x[i] =
(char*)malloc(20*sizeof(char));
        scanf("%s",x[i]);
    }
    inorder(n,x);
    for ( i = 0; i < n; i++)
```



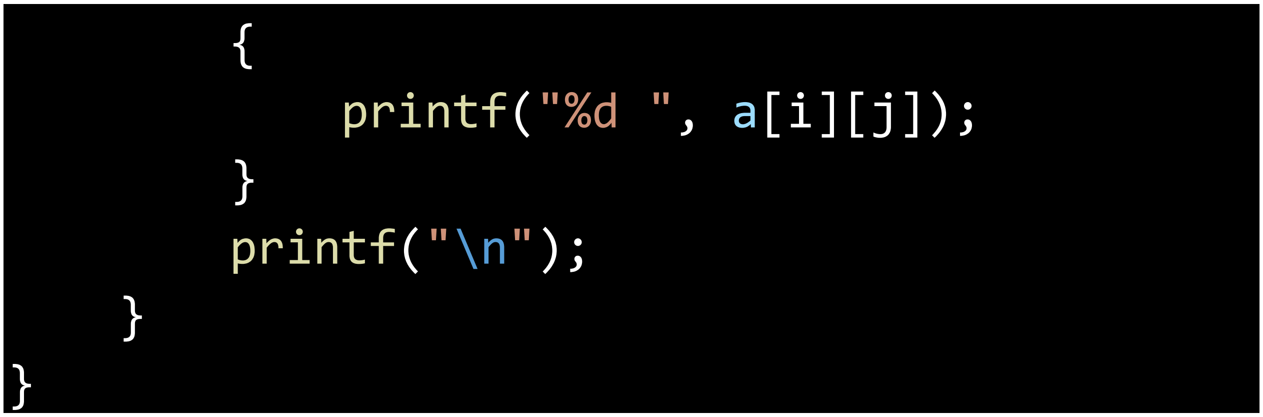
```
    {
        printf("%d %s\n",i+1,x[i]);
    }
    return 0;
}
void inorder(int n, char *x[])
{
    int i, j;
    char t[20];
    for ( i = 0; i < n-1; i++)
    {
        for (j = 0; j < n; j++)
        {
            if (strcmp(x[i],x[j])>0)
            {
                strcpy(t,x[j]);
                strcpy(x[j],x[i]);
                strcpy(x[i],t);
            }
            return;
        }
    }
}
/**
Name- Priayanshu Mallick, Roll No - 13
```



```
Q8. WAP to to input a matrix and display it
by using pointers.
The program should have two user defined
functions which pass
pointer argument for matrix.
**/
#include<stdio.h>
#include<stdlib.h>
void input(int m, int n, int **a);
void display(int m, int n, int **a);
int main()
{
    int m, n;
    printf("Enter numebr of rows and
column:\n");
    scanf("%d%d",&m,&n);
    int **a;
    a = (int**)malloc(m*sizeof(int*));
    for (int i = 0; i < m; i++)
    {
        a[i]=(int*)malloc(n*sizeof(int));
    }
    input(m, n, a);
    display(m, n, a);
    for (int i = 0; i < m; i++)
```



```
    {
        free(a[i]);
    }
    free(a);
    return 0;
}
void input(int m, int n, int **a)
{
    printf("Enter matrix elements:\n");
    int i,j;
    for ( i = 0; i < m; i++)
    {
        for ( j = 0; i < n; j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
}
void display(int m, int n, int **a)
{
    printf("matrix is:\n");
    int i,j;
    for ( i = 0; i < m; i++)
    {
        for ( j = 0; i < n; j++)
```



```
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
}
```