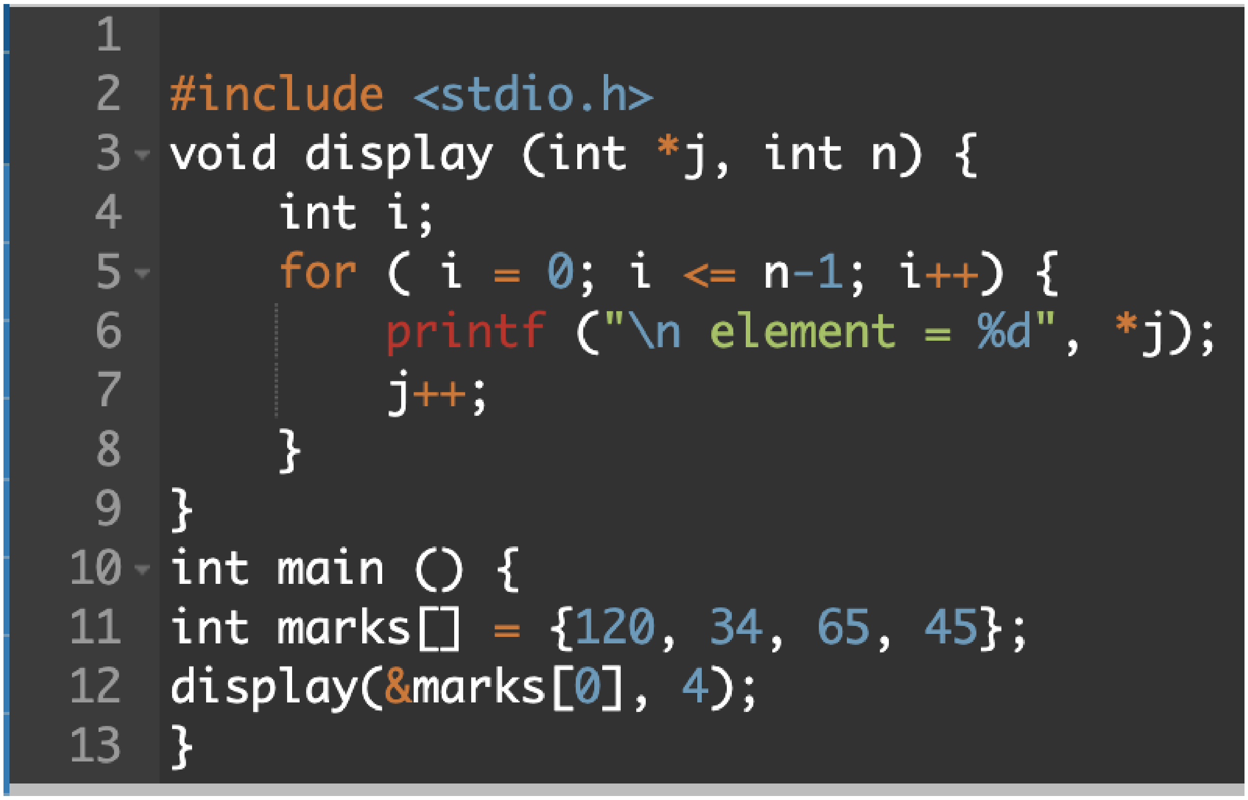
Birla Institute of Technology & Science, Pilani Hyderabad Campus

Second Semester 2020-2021

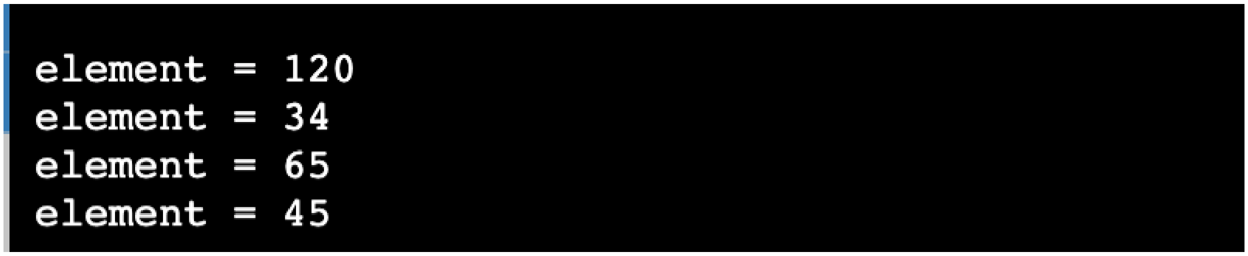
Computer Programming [CS F111]

Lab 10 (Arrays continued…)

Q1. Write a program to pass entire array elements to a function by using Call by Reference parameter passing mechanism.



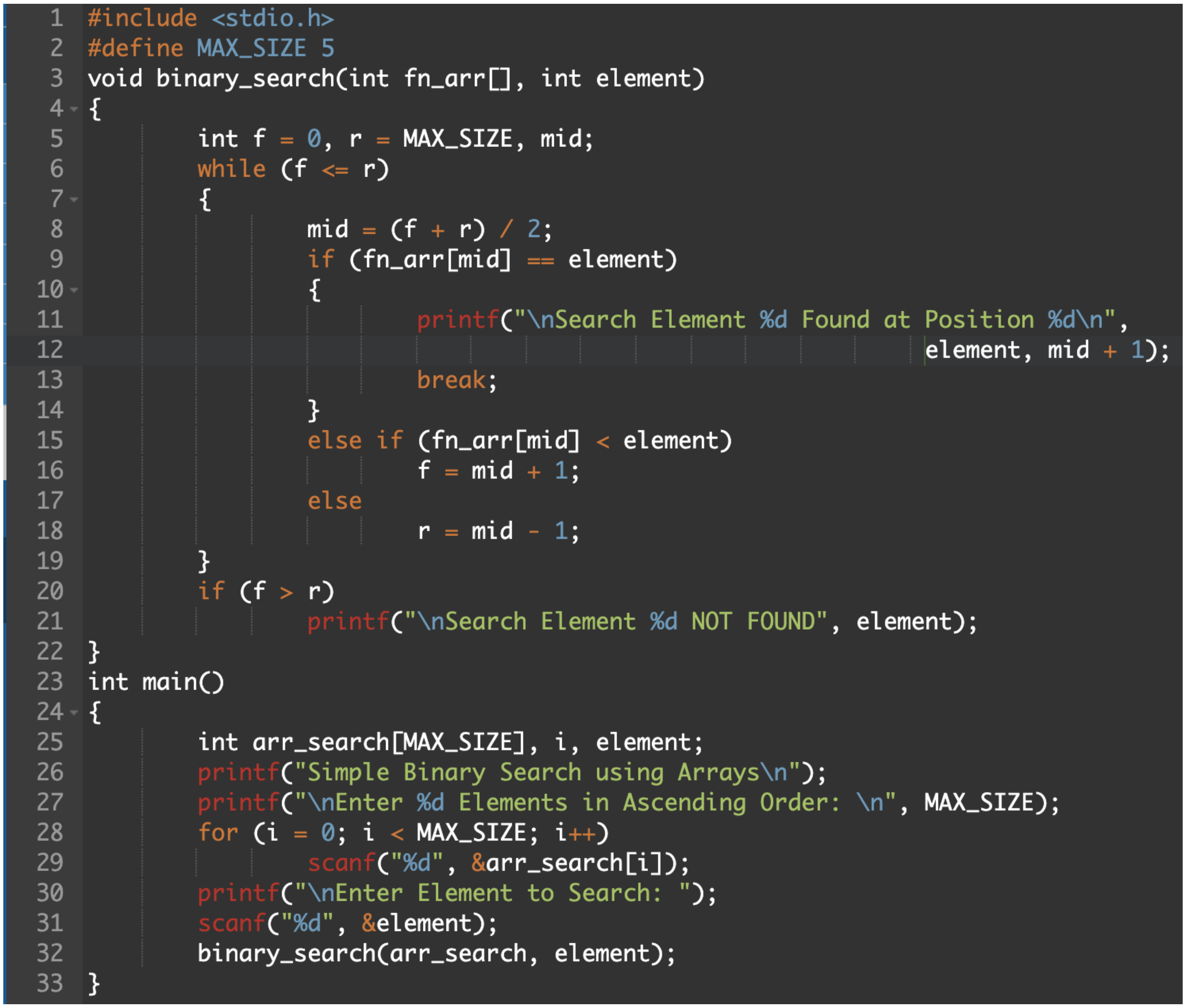
```

#include <stdio.h>
void display (int *j, int n) {
    int i;
    for ( i = 0; i <= n-1; i++) {
        printf ("\n element = %d", *j);
        j++;
    }
}
int main () {
int marks[] = {120, 34, 65, 45};
display(&marks[0], 4);
}
```



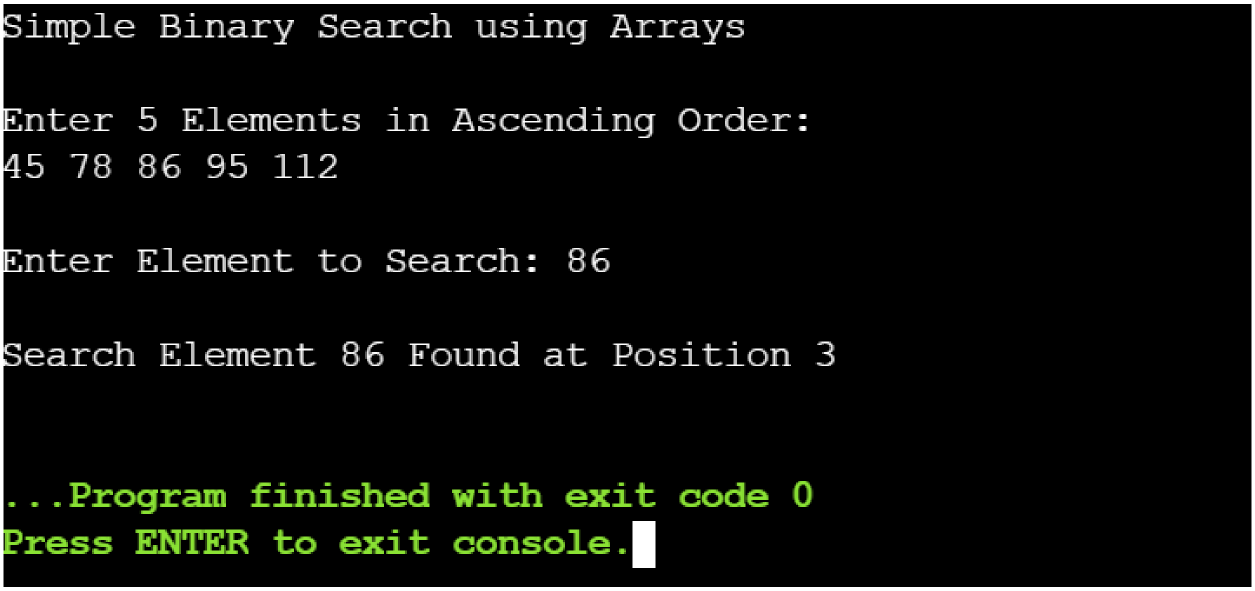
```
element = 120
element = 34
element = 65
element = 45
```

Task: Modify the above code to implement parameter passing mechanism using Call-by-Value.

Q.2 Write a program to search an element in an Array using **Binary search** technique.

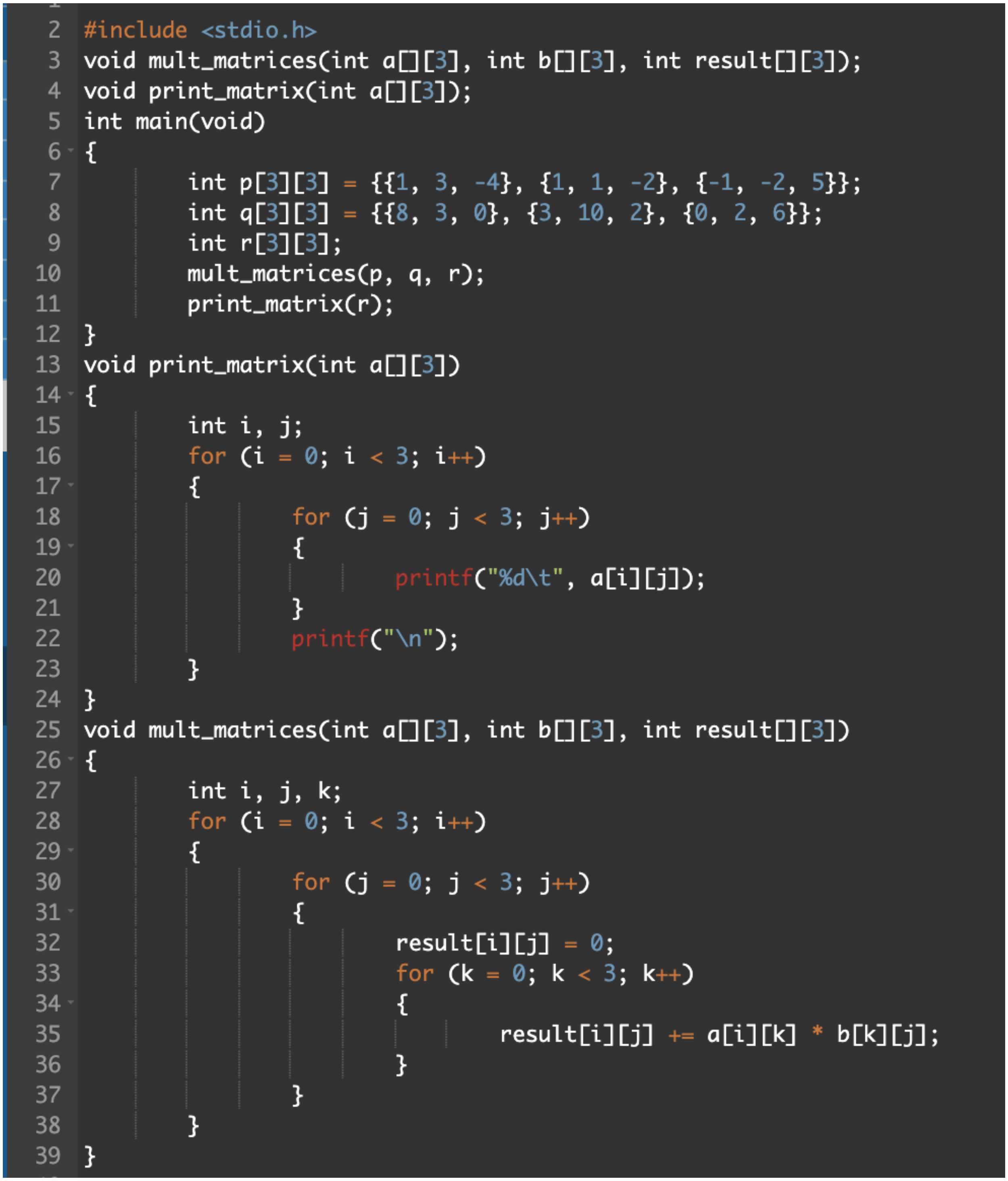


```c
#include <stdio.h>
#define MAX_SIZE 5
void binary_search(int fn_arr[], int element)
{
        int f = 0, r = MAX_SIZE, mid;
        while (f <= r)
        {
                mid = (f + r) / 2;
                if (fn_arr[mid] == element)
                {
                        printf("\nSearch Element %d Found at Position %d\n",
                                                                        element, mid + 1);
                        break;
                }
                else if (fn_arr[mid] < element)
                        f = mid + 1;
                else
                        r = mid - 1;
        }
        if (f > r)
                printf("\nSearch Element %d NOT FOUND", element);
}
int main()
{
        int arr_search[MAX_SIZE], i, element;
        printf("Simple Binary Search using Arrays\n");
        printf("\nEnter %d Elements in Ascending Order: \n", MAX_SIZE);
        for (i = 0; i < MAX_SIZE; i++)
                scanf("%d", &arr_search[i]);
        printf("\nEnter Element to Search: ");
        scanf("%d", &element);
        binary_search(arr_search, element);
}
```

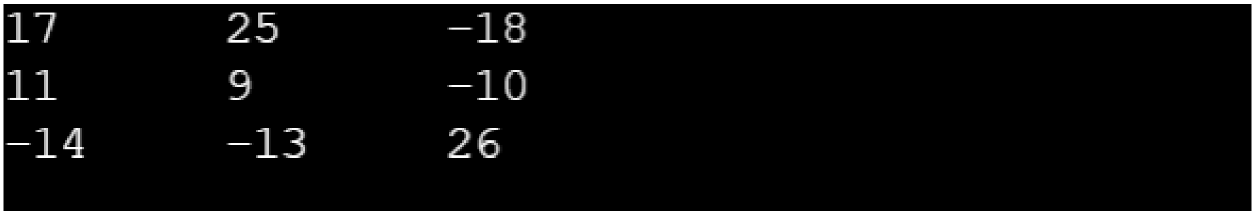


```
Simple Binary Search using Arrays

Enter 5 Elements in Ascending Order:
45 78 86 95 112

Enter Element to Search: 86

Search Element 86 Found at Position 3


...Program finished with exit code 0
Press ENTER to exit console.
```

Q.3 Write a program to multiply two matrices using Two-dimensional Arrays and print the resultant matrix.

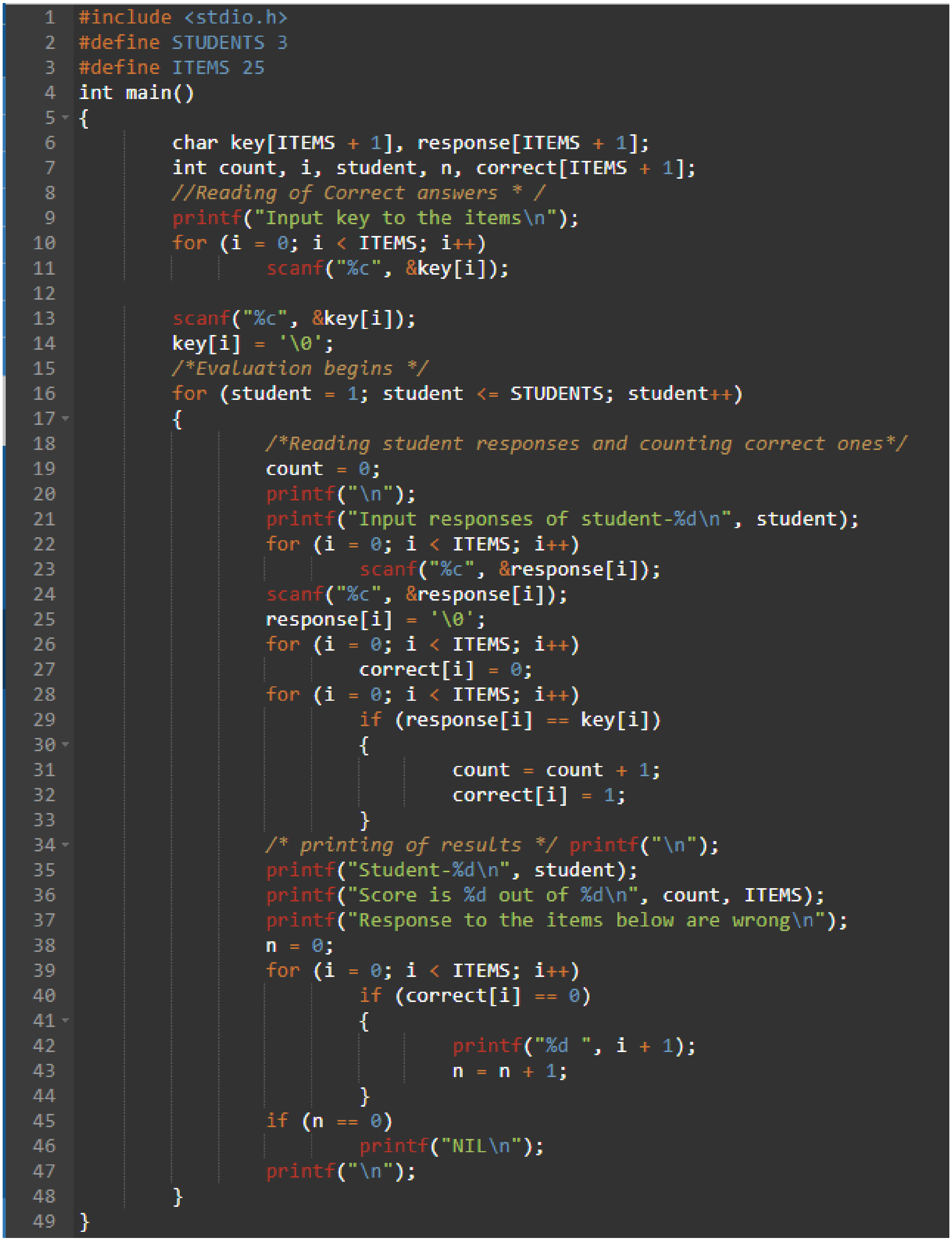


```c
#include <stdio.h>
void mult_matrices(int a[][3], int b[][3], int result[][3]);
void print_matrix(int a[][3]);
int main(void)
{
    int p[3][3] = {{1, 3, -4}, {1, 1, -2}, {-1, -2, 5}};
    int q[3][3] = {{8, 3, 0}, {3, 10, 2}, {0, 2, 6}};
    int r[3][3];
    mult_matrices(p, q, r);
    print_matrix(r);
}
void print_matrix(int a[][3])
{
    int i, j;
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }
}
void mult_matrices(int a[][3], int b[][3], int result[][3])
{
    int i, j, k;
    for (i = 0; i < 3; i++)
    {
        for (j = 0; j < 3; j++)
        {
            result[i][j] = 0;
            for (k = 0; k < 3; k++)
            {
                result[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}
```



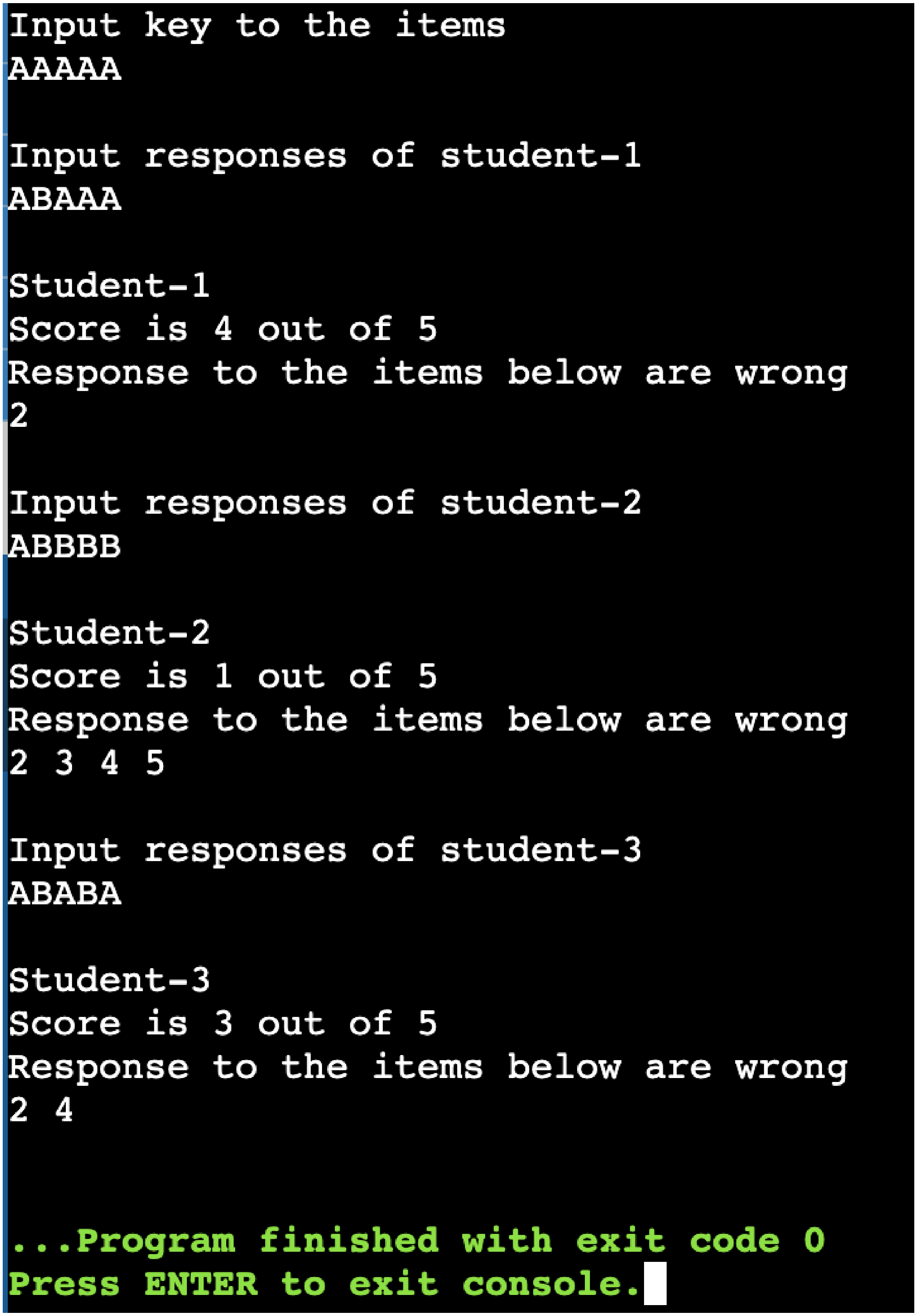
```
17      25      -18
11      9       -10
-14     -13     26
```

Q.4 Write a program to implement the Case study for evaluating answers in an Online Exam. Take input the Key, and the Responses of students. Display the scores of each student and the responses that are wrong.



```c
#include <stdio.h>
#define STUDENTS 3
#define ITEMS 25
int main()
{
        char key[ITEMS + 1], response[ITEMS + 1];
        int count, i, student, n, correct[ITEMS + 1];
        //Reading of Correct answers * /
        printf("Input key to the items\n");
        for (i = 0; i < ITEMS; i++)
                scanf("%c", &key[i]);

        scanf("%c", &key[i]);
        key[i] = '\0';
        /*Evaluation begins */
        for (student = 1; student <= STUDENTS; student++)
        {
                /*Reading student responses and counting correct ones*/
                count = 0;
                printf("\n");
                printf("Input responses of student-%d\n", student);
                for (i = 0; i < ITEMS; i++)
                        scanf("%c", &response[i]);
                scanf("%c", &response[i]);
                response[i] = '\0';
                for (i = 0; i < ITEMS; i++)
                        correct[i] = 0;
                for (i = 0; i < ITEMS; i++)
                        if (response[i] == key[i])
                        {
                                count = count + 1;
                                correct[i] = 1;
                        }
                /* printing of results */ printf("\n");
                printf("Student-%d\n", student);
                printf("Score is %d out of %d\n", count, ITEMS);
                printf("Response to the items below are wrong\n");
                n = 0;
                for (i = 0; i < ITEMS; i++)
                        if (correct[i] == 0)
                        {
                                printf("%d ", i + 1);
                                n = n + 1;
                        }
                if (n == 0)
                        printf("NIL\n");
                printf("\n");
        }
}
```

Output:



```
Input key to the items
AAAAA

Input responses of student-1
ABAAA

Student-1
Score is 4 out of 5
Response to the items below are wrong
2

Input responses of student-2
ABBBB

Student-2
Score is 1 out of 5
Response to the items below are wrong
2 3 4 5

Input responses of student-3
ABABA

Student-3
Score is 3 out of 5
Response to the items below are wrong
2 4


...Program finished with exit code 0
Press ENTER to exit console.
```

Task: Display the correct responses for each student too. ----------------