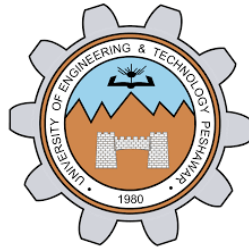


Operating Systems Lab-4
Introduction to C Programming



Submitted By: **Awais Saddiqui**

Registration# **21pwcse1993**

Section: **"A"**

Submitted to:

Mam Madiha Sher

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar.

CSE 302L: Operating Systems Lab

LAB ASSESSMENT RUBRICS

Marking Criteria	Exceeds expectation (2.5)	Meets expectation (1.5)	Does not meet expectation (0)	Score
1. Correctness	Program compiles (no errors and no warnings). Program always works correctly and meets the specification(s). Completed between 81-100% of the requirements.	Program compiles (no errors and some warnings). Some details of the program specification are violated, program functions incorrectly for some inputs. Completed between 41-80% of the requirements.	Program fails to or compile with lots of warnings. Program only functions correctly in very limited cases or not at all. Completed less than 40% of the requirements.	
2. Delivery	Delivered on time, and in correct format (disk, email, hard copy etc.)	Not delivered on time, or slightly incorrect format.	Not delivered on time or not in correct format.	
3. Coding Standards	Proper indentation, whitespace, line length, wrapping, comments and references.	Missing some of whitespace, line length, wrapping, comments or references.	Poor use of whitespace, line length, wrapping, comments and references.	
4. Presentation of document	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting and excellently organized.	Includes name, date, and assignment title. Task titles, objectives, output screenshots included and good formatting.	No name, date, or assignment title included. No task titles, no objectives, no output screenshots, poor formatting.	

Instructor:

Name: Engr. Madiha Sher

Signature: _____

C (programming language):

C programming language is a machine-independent programming language that is mainly used to create many types of applications and operating systems such as Windows, and other complicated programs such as the Oracle database, Git, Python interpreter, and games

Objectives:

- Writing simple C programs with more than one function
 - Basic concepts of Pointers in C
 - Using Arrays in C
 - Using Structures in C
 - Dynamic Memory Allocation
 - Use of Linked List
-

Task #1:

Write a program reads a number from user and finds its factorial using function. Pass the argument to function by value.

Code:

```
#include <stdio.h>

int factorial(int n);

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

    printf("Factorial of %d is %d\n", num, factorial(num));

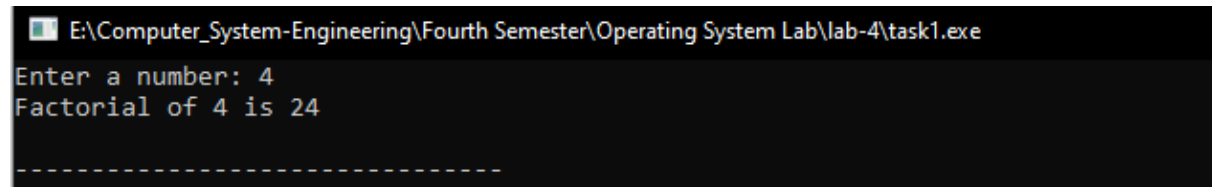
    return 0;
}

int factorial(int n) {
    int fact = 1;

    for (int i = 1; i <= n; i++) {
        fact *= i;
    }

    return fact;
}
```

Output:



```
E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\task1.exe
Enter a number: 4
Factorial of 4 is 24
-----
```

Task #2:

The following program demonstrates about the pointer variable, * and & operators. Run and observe the output.

Code:

```
#include <stdio.h>

int main (void) {

    int a;

    int *p;

    printf("Enter an Integer: ");

    scanf("%d",&a);

    p=&a;

    printf("The value of the variable a is %d\n",a);

    printf("The address of the variable a is %x\n",&a);

    printf("The value of variable p is %x\n",p);

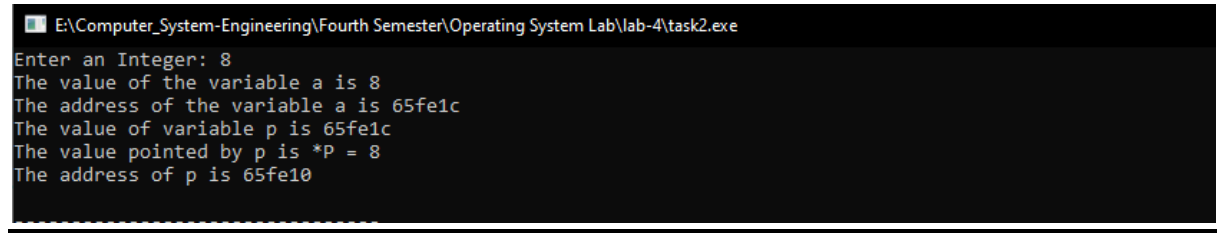
    printf("The value pointed by p is *P = %d\n",*p);

    printf("The address of p is %x\n",&p);

    return(0);

}
```

Output:



```
E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\task2.exe
Enter an Integer: 8
The value of the variable a is 8
The address of the variable a is 65fe1c
The value of variable p is 65fe1c
The value pointed by p is *P = 8
The address of p is 65fe10
```

Task #3:

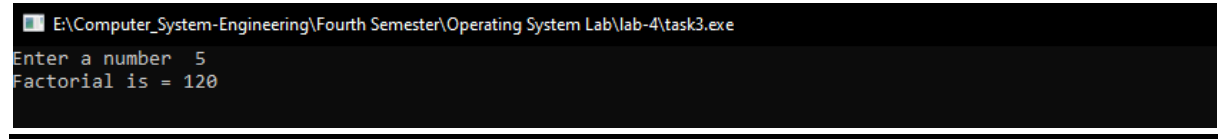
Redo task number 1. The result should be passed by pointer.

Code:

```
#include <stdio.h>
#include <stdlib.h>
int factorial(int *n){
    int fact=1;
    for(int i=1; i<=*n; i++){
        fact=fact*i;
    }
    return fact;
}
int main(void){
    int num;
    printf("Enter a number ");
    scanf("%d", &num);

    int result = factorial(&num);
    printf("Factorial is = %d\n",result);
}
```

Output:



```
E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\task3.exe
Enter a number 5
Factorial is = 120
```

Task #4:

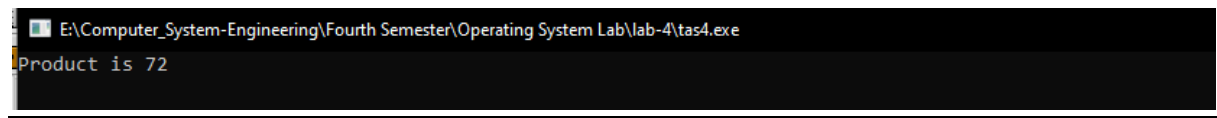
Write a function that calculates the dot product of two dimensional array. Call this function from main() function and display the product.

Code:

```
#include <stdio.h>
#include <stdlib.h>
int product(int arr[2][2]){
    int product=1;
    for(int i=0; i<2; i++){
        for(int j=0; j<2; j++){

            product =product*arr[i][j];
        }
    }
    return product;
}
int main(void){
    int arr[2][2]={{2,3},{3,4}};
    int result = product(arr);
    printf("Product is %d\n",result);
}
```

Output:



E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\tas4.exe
Product is 72

Task #5:

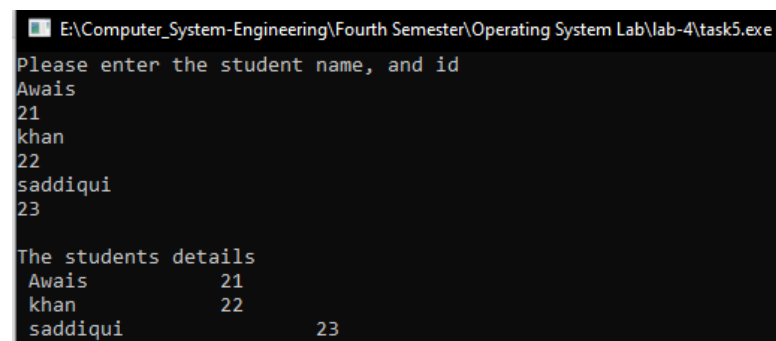
Run the following program and observe the output.

Code:

```
#include <stdio.h>

main()
{
    struct student
    {
        char name[20];
        int id;
    };
    struct student s1, s2, s3;
    printf("Please enter the student name, and id\n");
    scanf("%s %d", &s1.name, &s1.id);
    scanf("%s %d", &s2.name, &s2.id);
    scanf("%s %d", &s3.name, &s3.id);
    printf("\nThe students details");
    printf("\n %s \t\t %d", s1.name, s1.id);
    printf("\n %s \t\t %d", s2.name, s2.id);
    printf("\n %s \t\t %d", s3.name, s3.id);
}
```

Output:



```
E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\task5.exe
Please enter the student name, and id
Awais
21
khan
22
saddiqui
23

The students details
Awais      21
khan       22
saddiqui   23
```


Task #6:

Write a C code to declare "Time" structure that contains hour, minute and seconds as its data members. Write a function that adds two time instances and return the resultant time to the main function.

Code:

```
#include <stdio.h>

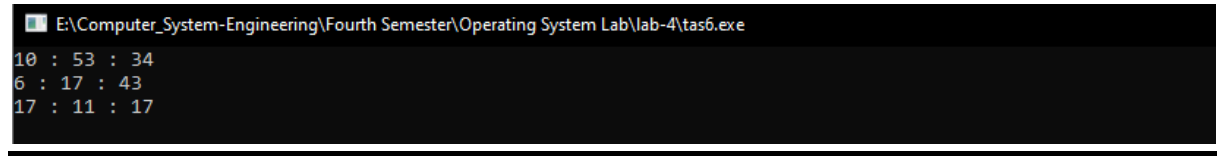
struct timeFun {
    int hour;
    int minute;
    int sec;
};

struct timeFun add_Time (struct timeFun t1, struct timeFun t2){
    struct timeFun t3;
    t3.hour = t1.hour + t2.hour;
    t3.minute = t1.minute + t2.minute;
    if (t3.minute >= 60){
        t3.hour++;
        t3.minute -= 60;
    }
    t3.sec = t1.sec + t2.sec;
    if (t3.sec >= 60){
        t3.minute++;
        t3.sec -= 60;
    }
    return t3;
}

void display_Time(struct timeFun t){
    printf("%d : %d : %d\n", t.hour, t.minute, t.sec);
}
```

```
int main(void){  
    struct timeFun t1, t2, t3;  
  
    t1.hour = 10;  
    t1.minute = 53;  
    t1.sec = 34;  
    display_Time(t1);  
  
    t2.hour = 6;  
    t2.minute = 17;  
    t2.sec = 43;  
    display_Time(t2);  
  
    t3 = add_Time(t1, t2);  
    display_Time(t3);  
    return 0;  
}
```

Output:



```
E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\tas6.exe  
10 : 53 : 34  
6 : 17 : 43  
17 : 11 : 17
```

Task #7:

Write a program that takes the size of the array as input from the user, create the array and then take the elements of array as input and sort in ascending order.

Code:

```
#include <stdio.h>

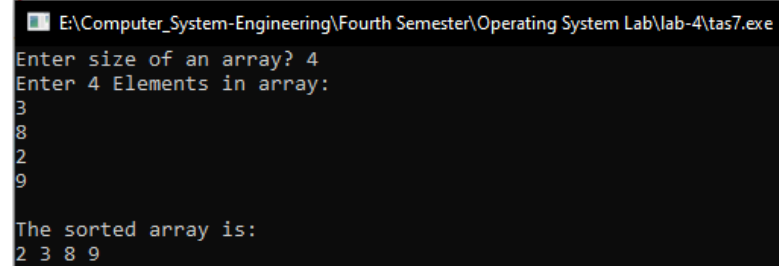
int* Sort(int arr[], int arraySize){
    int temp;
    for (int i = 0; i < arraySize; i++)
        for(int j = 0; j < arraySize-1; j++)
            if(arr[j] > arr[j + 1]){
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
    return arr;
}

void display(int arr[], int ofLength){
    for(int i = 0; i < ofLength; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

int main(void){
    int size;
    printf("Enter size of an array? ");
    scanf("%d", &size);
    int array[size];
    printf("Enter %d Elements in array: \n");
    for(int i = 0; i < size; i++)
        scanf("%d", &array[i]);
```

```
int sortedArray = Sort(array, size);  
printf("\nThe sorted array is: \n");  
display(sortedArray, size);  
  
return 0;  
}
```

Output:



The screenshot shows a Windows command prompt window with the title bar "E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\tas7.exe". The prompt displays the following text: "Enter size of an array? 4", "Enter 4 Elements in array:", followed by the input "3 8 2 9" on separate lines. The output of the program is "The sorted array is:" followed by "2 3 8 9" on the next line.

```
E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\tas7.exe  
Enter size of an array? 4  
Enter 4 Elements in array:  
3  
8  
2  
9  
  
The sorted array is:  
2 3 8 9
```

Task #8:

Write a complete menu driven program to do the following:

Code:

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

#define MAX_NAME_LENGTH 50

typedef struct node {
    char name[MAX_NAME_LENGTH];
    struct node *next;
} node;

typedef struct list {
    node *head;
    int size;
} list;

void init_list(list *l) {
    l->head = NULL;
    l->size = 0;
}

void add(list *l, char *newname) {
    node *temp = (node*) malloc(sizeof(node));
    strncpy(temp->name, newname, MAX_NAME_LENGTH);
    temp->next = l->head;
    l->head = temp;
    l->size++;
}
```

```

}

bool search(list *l, char *searchname) {
    node *temp = l->head;
    while (temp != NULL) {
        if (strncmp(temp->name, searchname, MAX_NAME_LENGTH) == 0) {
            return true;
        }
        temp = temp->next;
    }
    return false;
}

int main() {
    list l;
    init_list(&l);
    int choice;
    char name[MAX_NAME_LENGTH];

    while (true) {
        printf("1. Add name to list\n");
        printf("2. Search name in list\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1:
                printf("Enter the name to add: ");
                scanf("%s", name);
                add(&l, name);

```

```

        printf("Name added to the list!\n");
        break;
    case 2:
        printf("Enter the name to search: ");
        scanf("%s", name);
        if (search(&l, name))
            printf("Name found in the list!\n");
        else
            printf("Name not found in the list!\n");
        break;
    case 3:
        printf("Exiting...\n");
        exit(0);
    default:
        printf("Invalid choice!\n");
        break;
    }
}

return 0;
}

```

Output:

```

E:\Computer_System-Engineering\Fourth Semester\Operating System Lab\lab-4\tas8.exe
1. Add name to list
2. Search name in list
3. Exit
Enter your choice: 1
Enter the name to add: Awais
Name added to the list!
1. Add name to list
2. Search name in list
3. Exit
Enter your choice: 3
Exiting...

```