Tasks:

```c
void half_speed(){
    printf("Half speed\n");
}
void full_speed(){
    printf("Full speed\n");
}
void stop(){
    printf("Stop\n");
}
void main()
{
    char input;
    int count = 0;
    while(1){
        scanf(" %c",&input);
        if(input == 'c'){
            count++ ;
            switch(count){
                case 1 :
                    half_speed();
                    break;
                case 2 :
                    stop();
                    break;
                case 3 :
                    full_speed();
                    break;
                case 4 :
                    stop();
                    count=0;
                    break;
            }
        }
        else{
            printf("ERROR TRY AGAIN\n");
        }
    }
}
```

```
Windows PowerShell
PS C:\Users\hassa\OneDrive\□□□ □□□□□□□> gcc .\main.c
PS C:\Users\hassa\OneDrive\□□□ □□□□□□□> .\a.exe
c
Half speed
c
Stop
gc
Full speed
c
Stop
c
Half speed
d
ERROR TRY AGAIN
d
ERROR TRY AGAIN
c
Stop
c
Full speed
```

lab_7.c | lab_9.c | task_3.c | lab_11.c | main.c | lab_10.c

```c
#include <stdlib.h>
#include <stdint.h>
#include <string.h>
int *func(int*arr,int size){
    int i ,count_ones=0 ,count_negative=0;
    static int result[2];
    for(i=0 ; i < size ; i++){
        if(arr[i] == 1){
            count_ones++;
        }
        if(arr[i] < 0){
            count_negative++;
        }
    }
    result[0]=count_ones;
    result[1]=count_negative;
    return result;
}

void main(){
    int arr[10]={1,-5,-6,1,0,8,0,1,-8,-4};
    int *res;
    res = func(arr,10);
    printf("Number of ones = %d\n",*(res+0));
    printf("Number of negative = %d",*(res+1));
}
```

```
Number of ones = 3Number of negative = 4
PS C:\Users\hassa\OneDrive\□□□ □□□□□□□> gcc .\main.c
PS C:\Users\hassa\OneDrive\□□□ □□□□□□□> .\a.exe
Number of ones = 3
Number of negative = 4
PS C:\Users\hassa\OneDrive\□□□ □□□□□□□>
```

Labs

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

float * circle_calc(float num){
    static float calc [2];
    calc[0] = 3.14 * num * num;
    calc[1] = 2 * num *3.14;
    return calc;
}
/*
void main()
{

    float num,*calc;
    printf("Enter reduis : ");
    scanf("%f", &num);
    calc = circle_calc(num);
    printf("Area = %.2f \nCir = %.2f ",*calc,*(calc+1));
```

```
PS H:\NTI\C_language\labs> gcc .\lab_10.c
PS H:\NTI\C_language\labs> .\a.exe
Max = 30
Mini = -5
PS H:\NTI\C_language\labs>
```

```c
int * op_array(int *number){
    static int op [2];
    int i,j,temp ;
    for(i=0 ; i<5 ; i++){
        for(j = i+1 ; j <5 ; j++){
            if(*(number+j) > *(number+i)){
                temp = *(number+j);
                *(number+j) = *(number+i);
                *(number+i) = temp;
            }
        }
    }
    op[0] = *number;
    op[1] = *(number+4);
    return op;
}

void main()
{
    int number[5] = {10,30,-5,11,2};
    int *op = op_array(number);
    printf("Max = %d \nMini = %d ",*op,*(op+1));
```

```
PS H:\NTI\C_language\labs> gcc .
PS H:\NTI\C_language\labs> .\a.e
Max = 30
Mini = -5
```

```c
float add (float num_1 , float num_2){
    return (num_1+num_2);
}
float multi (float num_1 , float num_2){
    return (num_1*num_2);
}
float sub (float num_1 , float num_2){
    return (num_1-num_2);
}
float divid (float num_1 , float num_2){
    return (num_1/num_2);
}
void main(){
    printf("Enter first number : ");
    float num_1;
    scanf("%f",&num_1);
    printf("Enter operation : ");
    char op;
    scanf(" %c",&op);
    printf("Enter second number : ");
    float num_2;
    scanf("%f",&num_2);
    float (*ptr[4])(float , float)={add,multi,sub,divid};
    float result = 0;
    switch(op){
        case '+':
            result = ptr[0](num_1,num_2);
            printf("%.2f + %.2f = %.2f",num_1,num_2,result);
            break;
        case '*':
            result = ptr[1](num_1,num_2);
            printf("%.2f * %.2f = %.2f",num_1,num_2,result);
            break;
        case '-':
            result = ptr[2](num_1,num_2);
            printf("%.2f - %.2f = %.2f",num_1,num_2,result);
            break;
        case '/':
            result = ptr[3](num_1,num_2);
            printf("%.2f / %.2f = %.2f",num_1,num_2,result);
            break;
```

```
PS H:\NTI\C_language\labs>
PS H:\NTI\C_language\labs> .\a.exe
Enter first number : 5
Enter operation : -
Enter second number : 1
5.00 - 1.00 = 4.00
PS H:\NTI\C_language\labs> .\a.exe
Enter first number : 5
Enter operation : +
Enter second number : 3
5.00 + 3.00 = 8.00
PS H:\NTI\C_language\labs> .\a.exe
Enter first number : 6
Enter operation : *
Enter second number : 3
6.00 * 3.00 = 18.00
PS H:\NTI\C_language\labs>
PS H:\NTI\C_language\labs> .\a.exe
Enter first number : 6
Enter operation : /
Enter second number : 3
6.00 / 3.00 = 2.00
PS H:\NTI\C_language\labs>
```

```c
void op ( int (*ptr)[3] , int row , int col ){
    int i , j,max=ptr[0][0],mini=ptr[0][0];
    float avg=0;
    for (i=0 ; i < row ; i++){
        for(j=0 ; j < col ; j++){
            printf("%d\t", ptr[i][j]);

            if(ptr[i][j]>max){
                max = ptr[i][j];
            }

            if(ptr[i][j]<mini){
                mini = ptr[i][j];
            }

            avg = avg + ptr[i][j];
        }
        printf("\n");
    }
    printf("max number is %d\n", max);
    printf("mini number is %d\n", mini);
    printf("avarage number is %.2f\n", avg / 6);
}

void main(){
    int arra[2][3] = {{1,2,3},{4,5,6}};
    op(arra , 2 ,3);
}
```

```
PS H:\NTI\C_language\labs> gcc .\lab_11.c
PS H:\NTI\C_language\labs> .\a.exe
1       2       3
4       5       6
max number is 6
mini number is 1
avarage number is 3.50
PS H:\NTI\C_language\labs>
```

```c
int size_of_array(char *arr){
    int i ;
    for(i=0 ; i < 100 ; i++){
        if(arr[i] == '\0'){
            break;
        }
    }
    return i;
}

void main(){
    printf("Enter string : ");
    char arr[100];
    scanf("%s",arr);
    printf("size of string is : %d " , size_of_array(arr));

}
```

```
Enter string : hassan
size of string is : 6
```

```c
void main(){
    short num = 0x1234;
    void *p;
    p=&num;
    if (*(char*)p == 0x12){
        printf("big endian");
    }
    else{
        printf("little endian ");
    }
    //12 big endian ----- 34 little endian
}
```

```
Windows PowerShell
PS H:\NTI\C_language\labs> gcc .\lab_11.c
PS H:\NTI\C_language\labs> .\a.exe
little endian
PS H:\NTI\C_language\labs> gcc .\lab_10.c
```