

SOLENT UNIVERSITY

School of Media Arts and Technology

BSc (Hons) [Foundation]

Academic Year 2019-2020

Program solving

Contents

Task 1	4
Introduction	4
D Grade	7
C Grade	9
B Grade	11
A Grade	13
Task 2	16
Introduction	16
D Grade	20
C Grade	21
B Grade	23
A Grade	24
Task 3	26
Introduction	26
D Grade	28
C Grade	30
B Grade	33
A Grade	36
Task 4	39
D Grade	39
C Grade	41
B Grade	45
A Grade	49
Task 5	51
Introduction	51
D Grade	54
C Grade	56
B Grade	61
A Grade	65
Task 6	66
Introduction	66
D Grade	68
C Grade	71
B Grade	75
A Grade	77

Task 7	79
D Grade	79
C Grade	79
B Grade	81
A Grade	82
Task 8	83
Introduction.....	83
D Grade	86
C Grade	89
B Grade	93
A Grade	96
Reflective report.....	100

Task 1

Introduction

Concept of variables

A variable is a place in the memory which has a special name and special type. when we put a number into a variable then it can be changed easily by calling its name.

How we can define a variable?

Type of variable name[= initial amount of variable]

There are a few examples.

```
Int a;
```

```
Int a,b,c;
```

```
Int a=1,b=2,c;
```

```
Char chr;
```

```
Char ch2='x';
```

Where we can define a variable?

In C programming language we can define a new variable wherever we want. It can be defined either after main() or before while and for statement or wherever else.

How we can put initial amount in a variable?

There are two ways:

✓ When we define a variable, we can put initial amount

```
Int a=2;b;
```

```
Char chr1='c';
```

✓ 2.with scanf statement. With scanf statement the user who run our program can insert any valid number for variables.

```
Int a;
```

```
Scanf("%d", &a);
```

Scanf statement

This statement will be used for reading data from user.

`Scanf("%data type1,%data type 2 ...", &variable1,&variable2,...);`

Data types will be matched to variables. Depends on type of data can be used %c, %f, %d, %s.

- ✓ %c : for characters
- ✓ %d : for integers
- ✓ %f : for floating points
- ✓ % s : for strings

Examples:

Int num;

Scanf ("%d" , &num); “ we used %d because num is an integer variable“

Char ch1;

Scanf("%c", & ch1); “ we used %c because ch1 is a character variable“

Printf statement

This statement will be used for printing information. A program can be written without any input, but a program cannot be written without output.

`Printf("%data type1,%data type 2 ...", variable1,variable2,...);`

With printf statement also we can use \n or \t .

- \n : when a programmer wants to pointer jumps to the next line in output.
- \t: when a programmer wants to make gaps between outputs(Tab).

Having a user-friendly program

It is very important to print a message before asking something from a user. It helps to make a user-friendly program. For example:

`Printf("please insert first number: ");`

`Scanf("%d", &num1);`

So, it is necessary to print a message before getting it from user in order to have a user-friendly program.

If statement

If statement can be useful for conditions. Structure of if statement is very simple.

If (expression) statement

For more than one statement must use {}.

If (expression)

{

Statement 1

Statement 2

.

.

.

}

Also, if can be used with else:

If (expression) statement 1

Else statement 2

If the expression gets true then statement 1 will be run otherwise statement 2 will be run.

D Grade

```
main.c  saved
1  #include <stdio.h>
2
3  int main(void) {
4      int mileage, cost, deliveryfee;
5      printf("please insert mileage: ");
6      scanf("%d", & mileage);
7      if (mileage<=10) deliveryfee=0;
8      else if((mileage>10)&&(mileage<=20)) deliveryfee=10;
9      else if((mileage>20)&&(mileage<=30)) deliveryfee=15;
10     else if(mileage>30) deliveryfee=20;
11     if (cost==0) printf("your delivery fee is free");
12     else printf("\n your delivery fee is: %d ", deliveryfee);
13     return 0;
14 }
```

Fig1-task1-D Grade

To writing a program, first step is specifying inputs and outputs. In this task we need one variable as input:

- Mileage

To getting Mileage from user there are three steps:

- 1-difiniation of Mileage and its type which is integer (in line 4)
- 2-print a message for user before inserting mileage's amount to be more user friendly (in line 5).
- 3-reading mileage's amount from user with scanf command (in line 6).

For output we need only show delivery fee. So, I need to define another variable which is deliveryfee.

In each program we must do some process in our entries to having output. In this case we must do our processes in mileage. There are 4 option for mileage:

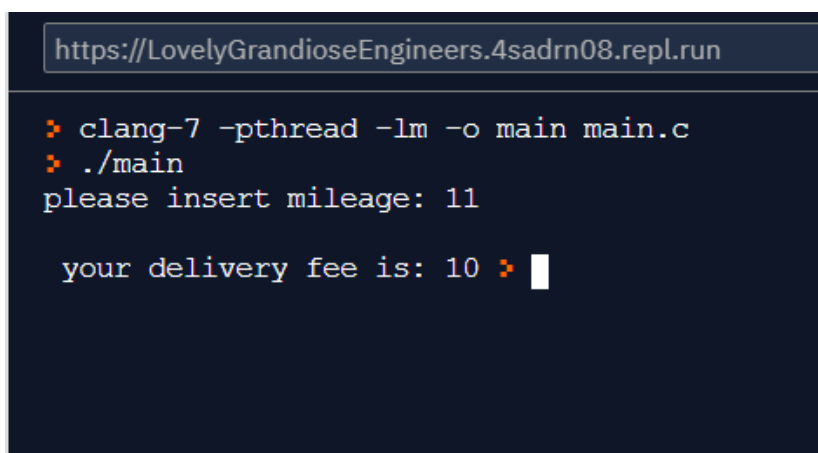
- 1- Delivery is free up to 10 miles: if mileage is less than 10 miles deliveryfee=0 (in line 7). If this condition gets true, then the program will go to the line 13 and it will not check the other conditions.
- 2- £10 over 10 miles: if the condition which is line 7 was not true then program will jump to line 8 and checks the condition in line 8. For checking an area between two numbers we must put 2 conditions with and between them.

3- £15 over 20 miles: in line number 9.

4- £20 over 30 miles: in line number 10.

Before program starts to run line 11, the last amount of deliveryfee has defined by one of our conditions. So just we need to add two conditions which if deliveryfee is equal 0 then print a message that your deliveryfee is free (line 11) otherwise print amount of delivery fee in separate line (line 12).

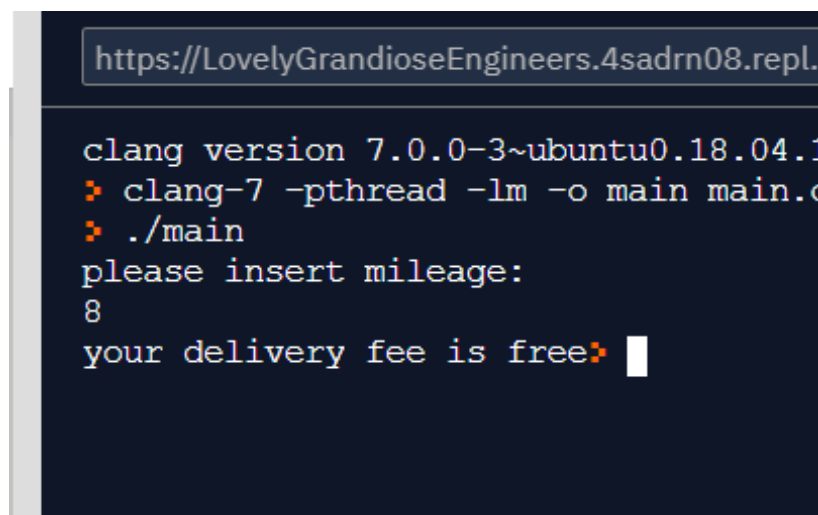
In fig2 I inserted 11 as mileage amount in output screen. So, the condition in line 8 got true and delivery fee is 10 £. I did the same in fig3, as you can see the condition in line 7 got true and the delivery fee got 0.

A terminal window with a dark background. The title bar shows the URL 'https://LovelyGrandioseEngineers.4sadrn08.repl.run'. The terminal content shows the compilation and execution of a C program. The user enters '11' for the mileage, and the program outputs 'your delivery fee is: 10' followed by a cursor.

```
https://LovelyGrandioseEngineers.4sadrn08.repl.run
❯ clang-7 -pthread -lm -o main main.c
❯ ./main
please insert mileage: 11

your delivery fee is: 10 ❯
```

Fig2-task1-D Grade

A terminal window with a dark background. The title bar shows the URL 'https://LovelyGrandioseEngineers.4sadrn08.repl.run'. The terminal content shows the compilation and execution of a C program. The user enters '8' for the mileage, and the program outputs 'your delivery fee is free' followed by a cursor.

```
https://LovelyGrandioseEngineers.4sadrn08.repl.run
clang version 7.0.0-3~ubuntu0.18.04.1
❯ clang-7 -pthread -lm -o main main.c
❯ ./main
please insert mileage:
8
your delivery fee is free❯
```

Fig3-task1-D Grade

C Grade

```
main.c  saved
1 #include <stdio.h>
2
3 int main(void) {
4     int mileage, cost, total, deliveryfee;
5     printf("please insert mileage: ");
6     scanf("%d", & mileage);
7     printf("please insert cost: ");
8     scanf("%d", & cost);
9     if (mileage<=10) deliveryfee=0;
10    else if((mileage>10)&&(mileage<=20)) deliveryfee=10;
11    else if((mileage>20)&&(mileage<=30)) deliveryfee=15;
12    else if(mileage>30) deliveryfee=20;
13    printf("\n you've paid: %d ", cost);
14    printf("\n You are [%d] miles away. ", mileage);
15    printf("\n the delivery fee is: %d ", deliveryfee);
16    printf("\n The total cost is[%d] ", cost+deliveryfee);
17
18
19    return 0;
20 }
```

Fig4-task1-C Grade

In this task we need some changes which are:

- 1- We need to get cost from user as well. So, we must define a new variable for cost (lines 4,7 and 8).
- 2- Instead of one output we have 4 outputs messages to showing to users (lines 13 to 16). We must print all information with user friendly format in output.

In line 16 I printed total cost which is add of deliveryfee and cost.

In fig 5 the user is 8 miles away. So, the condition in line 9 got true. In fig 6 the condition in line 10 got true as I inserted 12 as mileage's amount for entry.

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
please insert mileage: 8
please insert cost: 300

you've paid: 300
You are [8] miles away.
the delivery fee is: 0
The total cost is[300] ❖
```

Fig5-task1-C Grade

```
https://task1-grade-c.4sadrn08.repl.run

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
please insert mileage: 12
please insert cost: 120

you've paid: 120
You are [12] miles away.
the delivery fee is: 10
The total cost is[130] ❖
```

Fig6-task1-grade C

B Grade

```
1  #include <stdio.h>
2
3  int main(void) {
4  int mileage, cost, total, deliveryfee;
5      printf("please insert mileage: ");
6      scanf("%d", & mileage);
7      printf("please insert cost: ");
8      scanf("%d", & cost);
9      if((mileage<=10)&&(cost>=100))
10         deliveryfee=0;
11     else if((mileage<=10)&&(cost<100))
12         deliveryfee=5;
13     else if((mileage>10)&&(mileage<=20))
14         deliveryfee=10;
15     else if((mileage>20)&&(mileage<=30))
16         deliveryfee=15;
17     else if(mileage>30)
18         deliveryfee=15+((mileage-30)*0.5);
19     printf("\n you've paid: %d ", cost);
20     printf("\n You are [%d] miles away. ", mileage);
21     printf("\n the delivery fee is: %d ", deliveryfee);
22     printf("\n value of the goods is %d ", cost);
23     printf("\n The total cost is[%d] ", cost+deliveryfee);
24     return 0;
25 }
```

Fig7-task1-B Grade

For this task we must do some change to calculating deliveryfee as we have:

1-Delivery is free up to 10 miles if the value of the goods is over £100. So, in this case we must check amount of mileage and cost as well.

```
if((mileage<=10)&&(cost>=100))
```

```
    deliveryfee=0;
```

2- £5 up to 10 miles if the value of the goods is less than £100. In this case like first condition we must check mileage and cost as well.

```
if((mileage<=10)&&(cost>=100))
```

```
    deliveryfee=0;
```

3- £10 over 10 miles which we must check that the mileage is between 10 and 20.

```
else if((mileage>10)&&(mileage<=20))
```

```
    deliveryfee=10;
```

4- £15 over 20 miles

```
else if((mileage>20)&&(mileage<=30))
```

```
    deliveryfee=15;
```

5- £15 plus 50p per extra mile over 30 miles. In this case we must insert a formula.

deliveryfee=15+((mileage-30)*0.5);

for example, if mileage is 38, the deliveryfee is 15+((38-30)*0.5)=19

in fig 8 the user paid 300 £ which is 8 miles away. So the condition in line 9 got true. In fig 9 the user paid 70 £ which is 8 miles away, so the condition in line 11 got true. For fig 10 the condition in line 17 got true.

```
clang version 7.0.0-3~ubuntu0.18.04.1 (
> clang-7 -pthread -lm -o main main.c
> ./main
please insert mileage: 8
please insert cost: 300

you've paid: 300
You are [8] miles away.
the delivery fee is: 0
value of the goods is 300
The total cost is[300] >
```

Fig8-task1-B Grade

```
clang version 7.0.0-3~ubuntu0.18.0
> clang-7 -pthread -lm -o main ma
> ./main
please insert mileage: 8
please insert cost: 70

you've paid: 70
You are [8] miles away.
the delivery fee is: 5
value of the goods is 70
The total cost is[75] >
```

Fig9-task1-B Grade

```

clang version 7.0.0-3~ubuntu0.1
> clang-7 -pthread -lm -o main
> ./main
please insert mileage: 38
please insert cost: 300

you've paid: 300
You are [38] miles away.
the delivery fee is: 19
value of the goods is 300
The total cost is[319] > 

```

Fig10-task1-B Grade

A Grade

To doing this task we need to know how is syntax of switch case and when we can use it. Switch case can be used When there are many else if statements in a program. Syntax of switch case:

Switch (expression)

{

Case constant1: //statements

Break;

Case constant2: //statements

Break;

.

.

.

Default://statements

}

Next statement

The expression must be a character and all cases check constant values on expration. If expration in the first case is true then its statement will be run and after break statement, the next statement will be run. If the first case is not true

on expression then seconde case will be cheked. Compiler keep checking all cases until finds either a true expression or excute default statements. There is an example for switch case statement.

```
1  #include <stdio.h>
2
3  int main(void) {
4      int a,b;
5      char opt;
6      printf("please insert your first number: ") ;
7      scanf("%d",&a ) ;
8      printf(" please insert your second number: ") ;
9      scanf("%d",&b ) ;
10     printf("please insert your operator( + , -, *, / ): ");
11     scanf("%c", & opt);
12     switch (opt)
13     {
14         case '+': printf(" \n %d" , a+b );
15                 break;
16         case '-': printf(" \n %d - %d = %d ", a,b,a-b);
17                 break;
18         case '*': printf(" \n %d * %d = %d ", a,b,a*b);
19                 break;
20         case '/': { if (b!=0) printf("%d * %d = %d ", a,b,a/b);
21                   else printf(" \n error! division by zero!");
22                   break;
23                 }
24         default: printf(" \n error!! unkown operator!");
25     }
26     getchar();
27     return 0;
28 }
```

Fig11-task1-grade A

The program is design a simple calculator. According fig11, Entries are 2 numbers and type of operator. As it has been mantioned before the expration in front of switch must be a character. Opt is a character. If opt is '+' then the sum of the a and b wiil be indicated and after break statement, getchar() will be excuted. If opt is not '+' then next case will be checked. If opt is not '+ ,-* and / ' them default statement which is printing an error message will be excuted. In fig 13 is clear an error message.

```
https://TASK-1Greade-a1.4sadrn08.repl.run

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_7.0.0-3~ubuntu0.18.04.1)
> clang-7 -pthread -lm -o main main.c
> ./main
please insert your first number: 5
please insert your second number: 2
please insert your operator( + , -, *, / ): *

5 * 2 = 10 > 
```

Fig12-Task1-A Grade

```
https://TASK-1Greade-a1.4sadrn08.repl.run

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_7.0.0-3~ubuntu0.18.04.1)
> clang-7 -pthread -lm -o main main.c
> ./main
please insert your first number: 6
please insert your second number: 8
please insert your operator( + , -, *, / ): 0

error!! unkown operator!> 
```

Fig12-Task 1-A Grade

Task 2

Introduction

Concept of loops

Consider a programmer need to print integer numbers between 1 to 20. Without loops the programmer has to use 20 printf statements which is repetitive and unreasonable. Therefore using loop is a wise way to write this program. There are three loop statements in C language:

- While loop
- For loop
- Do..while loop

‘for’ statement can be used only for loops with a **certain repetition**. But while loop and do..while loop can be used for **both certain repetition and uncertain repetition**. Certain repetition means that programmer before create loop statements knows initial and final amount of counter.

Loop with determined repetition:

Each loop with certain repetition needs a **counter variable**. To create a loop, three **golden questions** must be answered:

1. What is initial amount of counter?
2. What is final amount of counter which is stopping condition for ending loop statement?
3. What is move step of counter? Is it increasing loop or decreasing loop?

After answering to **golden questions** is very simple to create a loop.

For example to printing integer number between 1 to 20:


1 (+1) 2 (+1) 319 (+1) 20



As it has been seen this is an increasing loop because initial amount is less than final amount ($1 < 20$). Also, the move step is +1. It means the counter must be added with one each time.

Another example is printing integer number between 20 to 1:

20 (-1) 19 (-1) 18 2 (-1) 1



As it has been seen this is a decreasing loop because initial amount is bigger than final amount ($20 > 1$). Also, the move step is -1. It means the counter must be reduced one each time. While loop structure:

```
Counter=initial amount;  
While (condition)  
{  
    .  
    .  
    .  
    Counter move step;  
}
```

To printing integer number between 1 to 20 :

```
Counter=1; //initial amount is 1  
While(counter<=20) //while loop will be executed until 20  
{  
    Printf("%d", &counter);  
    Counter++; //counter++ is equal with counter=counter+1  
}
```

How this while will execute? With each repetition of the loop expression will be checked if the expression is true then printf statement will be executed and after that counter++ will be executed. The loop will be finish when counter's amount is bigger than 20.

Step	counter	counter<=20	output
1	1	1<=20 true	1
2	2	2<=20 true	2
3	3	3<=20 true	3
.	.	.	.
.	.	.	.
20	20	20<=20 true	20
21	21	21>20 false	-----

Using &&, ||, !=, == in expressions:

Sometimes it is needed to check more than one value in an expression in front of a while loop or even an if statement. depends on condition “AND, OR, NOT” can be used. For equality == should be used. For example:

```

If (counter==20)
{
    Printf("yes");
}
Printf("no");

```

If counter is equal with 20 then yes otherwise no will be printed. Let us discuss about AND. True table for AND is:

A	B	A && B
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

If both A and B are TRUE, then output is TRUE. Otherwise if either one of A and B OR both are false then output is FALSE.

So, when a programmer needs to check either for a loop or if more than one expression then AND can be used. Or even for checking a range on a variable can be used. For example, if it is needed to check that a counter is between 1 and 20:

```

If ( (counter>=1)&&(counter<=20))
{
Printf("yes");
}
Printf("no");

```

If the counter is greater than 1 and less than 20 'yes' will be show otherwise 'no' will be show. For example, if counter is equal 15 then expression is true and 'yes' will be show.

In 'OR', if just one of operands is true then output gets true.

A	B	A B
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

For example:

```

If ((Counter==10) || (counter==20))
{
Printf("yes");
}
Printf("no");

```

if counter is equal 10, then one of expressions is true and the output is yes. if counter is equal 20, then again one of expressions is true and the output is yes. But if for example, counter is equal 8 then both expressions are false and no will be show in output.

Sometimes need to check opposite a value. Then != can be used. For example:

```

If (opt != 'n')
{
i++;
}
i--;

```

if opt is not equal 'n' then i++ otherwise i-- will be executed.

D Grade


```
main.c  saved
1  #include <stdio.h>
2
3  int main(void) {
4      int a, i;
5      printf("hello!");
6      printf("\n please insert an integer number between 1 to 20 : ");
7      scanf("%d", &a);
8      i=1;
9      while(i<=20)
10     {
11         if (i!=a) printf("_ ");
12         else printf(" x ");
13         i++;
14     }
15     getchar();
16     return 0;
17 }
```

Fig13-TASK2-D GRADE

This program displays 20 dash on the screen. The program asks the user for a number between 1 and 20. This number will represent the location of the X. it will print out an 'X' instead of a dash in location of the entry number from user.

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
hello!
please insert an integer number between 1 to 20 : 3
_ _ X _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ ❖
```

Fig14-TASK2-D GRADE

C Grade

```
main.c  saved
1  #include <stdio.h>
2
3  int main(void) {
4      int a, i;
5      printf("hello!");
6      printf("\n please insert an integer number between 1 to 20 : ");
7      scanf("%d", &a);
8      if ((a>=1)&&(a<=20))
9      {
10         i=1;
11         while (i<=20)
12         {
13             if (i==a) printf(" x ");
14             else printf(" _ ");
15             i++;
16         }
17     }
18     else printf("sorry! you entered incorrect number! bye");
19
20     return 0;
21 }
```

Fig15-taske2-C Grade

In this task if the user enters a number that is not between 1 and 20, the program should not display any dashes and instead of dashes will print an error message "Sorry, the number needs to be between 1 and 20".

Line 5 to 7 is printing a message for user and then getting a number between 1 to 20 from user. This number goes to 'a' variable.

To checking the range of 1 to 20 for a variable is needed to have an if statement. If 'a' is b between 1 and 20 then the program goes to dashes and x printing area otherwise without printing any dashes, after an error message program will be ended.

```
scanf("%d", &a);
if ((a>=1)&&(a<=20)) //checking range of 1 to 20 for 'a' variable
{
    Dashes and x printing area
}
Else
printf("sorry! you entered incorrect number! bye");//printing error message
```

The counter in this program is 'i' and the entry number from user is 'a'. To print dashes the programmer needs a loop between 1 to 20 and each time counter will be added one.

--

```
if (i!=a) printf(" _ ");  
else printf(" X ");
```

```
https://TASK-2-GRADE-C.4sadrn08.repl.run
```

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
hello!
please insert an integer number between 1 to 20 : 6
- - - - - X - - - - - ❖ □
```

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
➤ clang-7 -pthread -lm -o main main.c
➤ ./main
hello!
  please insert an integer number between 1 to 20 : 25
sorry! you intered incorrect number! bye➤
```

Fig17-taske2-C Grade

B Grade

```
main.c  saved
1  #include <stdio.h>
2
3  int main(void) {
4      int a,distance, i;
5      printf("hello!");
6      printf("\n please insert an integer number between 1 to 20 : ");
7      scanf("%d", &a);
8      printf("Please enter the distance between the two X's ");
9      scanf("%d", &distance);
10     if ((a>=1)&&(a<=20))
11     {
12         i=1;
13         while (i<=20)
14         {
15
16             for(i=1;i<=20; i++)
17                 if ((i==a)|| (i==(a+distance))) printf(" X ");
18                 else printf(" _ ");
19         }
20         i++;
21     }
22     else printf("sorry! you entered incorrect number! bye");
23
24     return 0;
25 }
```

Fig18-taske2-B Grade

For this task need to get two inputs from user. One for the location where they want the first 'X' to appear, and the second input will represent the location of a second 'X'. specifically, it will represent the distance the second 'X' will be from the first one. For example, if the user entered 3 for the first X and then 5 for the distance. The first X will be printed in location 3 and the second X will be printed 5 dashes away from it where is in location number 8 as $3+5=8$. So, for this task just need to change if statement in the bellow of 'for' statement.

The first number called 'a' and the second number is 'distance'. Consider $a=3$ and $distance=7$, then the location of first 'X' is 3rd and the location of second 'X' is 10 as $(a+distance=10)$. So, if 'i' is equal 'a' or 'i' is equal 'a+distance' instead of printing a dash must be printed a 'X'. therefore, the only statement which must be changed is if statement.

```
if ((i==a) || (i==(a+distance))) printf(" X ");
else printf(" _ ");
```

```
https://TASK-2-GRADE-B.4sadrn08.repl.run

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang-7 -pthread -lm -o main main.c
> ./main
hello!
please insert an integer number between 1 to 20 : 6
Please enter the distance between the two X's 4
- - - - - X - - - - - X - - - - -
```

Fig19-taske2-B Grade

A Grade

To doing this task is needed to look at 'for' statement. as it has been maintained in concepts of loop's section, 'for' statement can be used only for loops with a **certain repetition**. Like while statement here is needed to ask yourself three **golden questions**.

'For' loop structure:

For(counter=initial amount;checking final amount; move step)

Stetement;

If there is more than one statement which 'for' must excute them with each time repeat then is needed to put them inside of { }, otherwise 'for' just can see first statement with each repeat and it cause to change the result of program. For example:

Printing integer number between 1 to 20.

For (int i=1; i<=20; i++)

Printf("%d", i);

Initial amount is 1, as here the first number is 1. To chacking final amount is need to check each time which does counter get 20 or no. so, the final amount is getting 20. Final amount can be written 'i<=20 ' or 'i<21'. Let us to print just odd numbers between 1 to 100. First odd number in this range is 1. So, the initial amount is 1. The last odd number between 1 to 100 is 99. So, final amount for counter is 99. Let us talk about move atep. Odd numbers between 1 to 99 are 1,3,5,7,9,...,97,99. So move step is two because between each number there is distance of two. The 'for' statement will be:

For(int counter=1;counter<=99; counter+=2)


```
Printf("%d", counter);
```

```
//Counter+=2 is equal counter=counter+2
```

Now let us talk about how it can be changed a written program with while loop to a program with for loop. It is really simple. Just is needed to know the answer of those three **golden questions** and also know that what is doing the while loop with each time repeat.

The task in B Grade will be written by for loop. First of all must be finded the answers of those three **golden questions** .

```
i=1;//initial amount
while (i<=20)//final amount
{

    if ((i==a) || (i==(a+distance))) printf(" X ");
    else printf(" _ ");

    i++;//move step
}
```

As it can be seen it is really simple to find the answers of **golden questions**. If we delete whatever have been finded then clearly can be seen that the while loop with each time repeat excutes the if statements:

```
for(i=1;i<=20; i++)
    if ((i==a) || (i==(a+distance))) printf(" X ");
    else printf(" _ ");
```

```
int main(void) {
    int a,distance, i;
    printf("hello!");
    printf("\n please insert an integer number between 1 to 20 : ");
    scanf("%d", &a);
    printf("Please enter the distance between the two x's ");
    scanf("%d", &distance);
    if ((a>=1)&&(a<=20))
    {
        for(i=1;i<=20; i++)
            if ((i==a) || (i==(a+distance))) printf(" X ");
            else printf(" _ ");
    }
    else printf("sorry! you intered incorrect number! bye");
    return 0;
}
```

Fig20-taske2-A Grade

```
clang version 7.0.0-3-ubuntu0.18.04.1 (tags/RELEASE_700/final)
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
hello!
  please insert an integer number between 1 to 20 : 6
Please enter the distance between the two X's 3
_ _ _ _ _ X _ _ _ X _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ ❖
```

Fig21-taske2-A Grade

Task 3

Introduction

Independent nested loops:

When there are two continuous loop and their counters are not related they are independent nested loops. The best example for that is a multiplication table.

1*1=1	1*2=2	1*3=3	1*10=10
2*1=2	2*2=4	2*3=6	2*10=20
.....
10*1=10	10*2=20	10*3=30		10*10=100

As it is clear first operands in each row are fixed while second operands are changing. Therefore two loops are needed. One loop is for first operands and the other one for second operands.

```
for(int counter1=1; counter1<=10; counter1++)
    for(int counter2=1; counter2<=10; counter2++)
        printf("%d", conter1*counter2);
```

when counter1 is 1, counter2 is counting from 1 to 10. When counter2 changes to 11 then printf statement doesn't execute and program again goes to first loop and counter2 is added one. This procedure will be finished when counter2=11.

counter1	counter2	counter1*counter2(output)
1	1..10	1,2,3...9,10
2	1..10	2,4,6...18,20
.....
10	1..10	10,20,30..100

Loop with indeterminate repetition:

To making an interminate loop can use either **while** or **do..while** statements. There is not any counter for making indeterminate loop as it is not clear total time of loop's repeat. Here is needed to use a character variable with initial amount. For exapmle considere flag=0 before starting a loop. Loop finishes when flag=1. Each time shuld be asked from the user that "do you want to continue?". if the user wants to continue then flag must be changed to 1 otherwise the flag dosen't change. For example getting numbers from the user and the output is sum of numbers. The programmer dosen't know the number of entries.

```
Int num,sum=0;
```

```
Char chr;
```

```
Int flag=0; //intial amount for flag
```

```
While(flag!=1) //ending condition is when flag=1
```

```
{
```

```
    Printf("please insert a number: ");
```

```
    Scanf("%d",&num );
```

```
    Sum=sum+num; // as soos as gets a number, will add to sum
```

```
    Printf("do you want to continue? y/n");
```

```
    Scanf("%c", &chr);//the user will insert y or n
```

```
    If (chr=='n') flag=1; //just if the user insrt n,the flag changes
```

```
}
```

```
Printf ("%d", &sum);// when the flag=1 then while condition gets true and and the program jumps to the statement after braket which is printf statement.
```

D Grade

```
int main(void) {
    char opt;
    int num1,num2,i,j;
    float r;
    printf("Please enter the letter which corresponds with your choice");
    printf("\n a - Calculate the area of a rectangle");
    printf("\n b - Calculate the area of a circle");
    printf("\n c - Display a multiplication table");
    printf("\n d - Add two numbers : " );
    scanf("%c", &opt);

    switch (opt)
    {
        case 'a': {
            printf("You have selected a -calculate the area of a rectangle" );
            printf("\n Please enter the width ");
            scanf("%d", &num1);
            printf("\n Please enter the width ");
            scanf("%d", &num2);
            printf("your result is: %d ", num1*num2);
            break;
        }
        case 'b':{
            printf(" You have selected b -Calculate the area of a circle");
            printf("\n Please enter the radius ");
            scanf("%f",&r);
            printf("your result is: %f ", r*r*3.14);
            break;
        }
    }
}
```

Fig22-taske3-D Grade

```
        case 'c': {
            printf("You have selected b -Display a multiplication table\n");
            for(i=1;i<=10;i++)
            {
                for(j=1;j<=10;j++)
                {
                    printf("%4d", i*j);
                    printf("\n");
                }
            }
            break;
        }
        case 'd': {
            printf("You have selected b -Display a multiplication table\n");
            printf("\n Please enter the first number ");
            scanf("%d", &num1);
            printf("\n Please enter the first number ");
            scanf("%d", &num2);
            printf("%d + %d =%4d", num1,num2,num1+num2);
            break;
        }
        default: printf(" \n error!! unkown operator!");
    }
    getchar();
    return 0;
}
```

Fig23-taske3-D Grade

```

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang-7 -pthread -lm -o main main.c
> ./main
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers : a
You have selected a -calculate the area of a rectangle
Please enter the width 2

Please enter the width 3
your result is: 6 >

```

Fig24-taske3-A Grade

```

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang-7 -pthread -lm -o main main.c
> ./main
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers : c
You have selected b -Display a multiplication table
 1  2  3  4  5  6  7  8  9 10
 2  4  6  8 10 12 14 16 18 20
 3  6  9 12 15 18 21 24 27 30
 4  8 12 16 20 24 28 32 36 40
 5 10 15 20 25 30 35 40 45 50
 6 12 18 24 30 36 42 48 54 60
 7 14 21 28 35 42 49 56 63 70
 8 16 24 32 40 48 56 64 72 80
 9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
>

```

Fig25-taske3-D Grade

```

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang-7 -pthread -lm -o main main.c
> ./main
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers : b
You have selected b -Calculate the area of a circle
Please enter the radius 3
your result is: 28.260000 >

```

Fig26-taske3-D Grade

C Grade

```
#include <stdio.h>

int main(void) {
    char opt;
    int num1,num2,i,j;
    float r;
    printf("Please enter the letter which corresponds with your choice");
    printf("\n a - Calculate the area of a rectangle");
    printf("\n b - Calculate the area of a circle");
    printf("\n c - Display a multiplication table");
    printf("\n d - Add two numbers " );
    scanf(" %c", &opt);
    switch (opt)
    {
        case 'a':
            printf("\n Please enter the width ");
            scanf("%d", &num1);
            printf("\n Please enter the width ");
            scanf("%d", &num2);
            printf("your result is: %d ", num1*num2);
            break;
        case 'b':
            printf("\n Please enter the radius ");
            scanf("%f",&r);
            printf("your result is: %f ", r*r*3.14);
            break;
        case 'c':
            for (i=1; i<=10; i++)
            {
                for (j=1; j<=10; j++)
                {
                    printf("%d\t", i*j);
                    if (j%10 == 0)
                        printf("\n");
                }
            }
            break;
        case 'd':
            printf("\n Please enter the first number ");
            scanf("%d", &num1);
            printf("\n Please enter the second number ");
            scanf("%d", &num2);
            printf("your result is: %d ", num1+num2);
            break;
    }
}
```

Fig27-taske3-C Grade

```
case 'A':
case 'a': {printf("You have selected a -calcculate the area of a rectangle" );
            printf("\n Please enter the width ");
            scanf("%d" , &num1);
            printf("\n Please enter the width ");
            scanf("%d" , &num2);
            printf("your result is: %d ", num1*num2);
            break;}
case 'B':
case 'b':{ printf(" You have selected b -Calculate the area of a circle");
            printf("\n Please enter the radius ");
            scanf("%f",&r);
            printf("your result is: %f ", r*r*3.14);
            break;
        }
}
```

Fig28-taske3-C Grade

```

case 'C':
case 'c': { printf("You have selected C -Display a multiplication
table\n");
for(i=1;i<=10;i++){
for(j=1;j<=10;j++){
printf("%4d", i*j);
printf("\n");
}
break;
}
case 'D':
case 'd': {printf("You have selected D- Add two numbers\n");
printf("\n Please enter the first number ");
scanf("%d" , &num1);
printf("\n Please enter the first number ");
scanf("%d" , &num2);
printf("%d + %d =%4d", num1,num2,num1+num2);
break;
}
default: printf("Sorry, your input was not recognised, please
enter either a, b, c or d");
}
getchar();
return 0;
}

```

Fig29-taske3-C Grade

This task is creating a menu with four options which each option can does different job. Depending on what letter the user entered, the program should perform the appropriate action. The program should inform the user if they entered an invalid menu choice . The program should accept both upper-case and lower-case letters. There are three steps to creating a menu:

1. Printing menu's options: it is simple to print menu's option as a message with printf statement.
2. Getting user's choice: user's entry goes to 'opt' which is a character.
3. Implementation : implementation is with switch case statement.

All cases in switch case statement checks with 'opt'. if a matching option is found then program executes matching option's statement and then after break statement the program jumps to getchar(). If there is not any matching option then the program executes default's statement. To accepting both upper-case and lower-case letters should put both as continuous cases in switch case statement:

```

case 'A':
case 'a':

```

The first option is Calculating the area of a rectangle. The entries are num1 and num2. The second option is Calculating the area of a circle. The entry is 'r' which is radius of the circle. The third option is displaying a multiplication table which

is shown in requirement section. To moving cursor to next line `printf("\n")` is used. When second loop ends, the next statement is `printf` statement as after first loop open bracket is used and this bracket is closed after `printf` statement. it means second loop and `printf` statements are in first loop's area. After `printf`, again first loop executes. Each `for` statement can execute just one statement with each time repeat. If is needed to execute more than one statement must put those statements in brackets.

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang-7 -pthread -lm -o main main.c
> ./main
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers D
You have selected D- Add two numbers

Please enter the first number 2

Please enter the first number 3
2 + 3 = 5
```

Fig30-taske3-C Grade

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang-7 -pthread -lm -o main main.c
> ./main
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers d
You have selected D- Add two numbers

Please enter the first number 4

Please enter the first number 5
4 + 5 = 9
```

Fig31-taske3-C Grade

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang-7 -pthread -lm -o main main.c
> ./main
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers l
Sorry, your input was not recognised, please enter either a, b, c or d
```

Fig32-taske3-C Grade

B Grade

```
#include <stdio.h>

int main(void) {
    char opt='a';
    int num1,num2,i,j;
    float r;
    while(opt!='x')
    {
        printf("Please enter the letter which corresponds with your
        choice");
        printf("\n a - Calculate the area of a rectangle");
        printf("\n b - Calculate the area of a circle");
        printf("\n c - Display a multiplication table");
        printf(" \n d - Add two numbers ");
        printf(" \n X -exit ");
        printf("\n please insert your option: ");
        scanf(" %c", &opt);

        switch (opt)
        {
            case 'A':
            case 'a': {printf(" You have selected a -calculate the area of a
            rectangle" );
            printf
            ("-----\n");
            printf("\n Please enter the length ");
            scanf("%d" , &num1);
            printf("\n Please enter the width ");
            scanf("%d" , &num2);
            printf("your result is: %d ", num1*num2);
            break;}
        }
```

Fig33-taske3-B Grade

```
        case 'B':
        case 'b': { printf(" You have selected b -Calculate the area of a
        circle");
        printf
        ("-----\n");
        printf("\n Please enter the radius ");
        scanf("%f",&r);
        printf("your result is: %f ", r*r*3.14);
        break;
        }
        case 'C':
        case 'c': { printf("You have selected b -Display a multiplication
        table\n");
        printf
        ("-----\n");
        for(i=1;i<=10;i++){
            for(j=1;j<=10;j++)
                printf("%4d", i*j);
            printf("\n");
        }
        break;
        }
        case 'D':
        case 'd': {printf("You have selected C-Add two numbers\n");
        printf
        ("-----\n");
        printf("\n Please enter the first number ");
        scanf("%d" , &num1);
        printf("\n Please enter the first number ");
        scanf("%d" , &num2);
        printf("%d + %d =%4d", num1,num2,num1+num2);
        break;
        }
```

Fig34-taske3-B Grade

```

    }
    case 'x' :
    case 'X' : {printf("end of program! \n");
    printf
    ("-----\n");
    opt='x';
    break;
    }
    default: printf("Sorry, your input was not recognised, please
    enter either a, b, c or d");
    }
    }
    getchar();
    return 0;
}

```

Fig35-taske3-B Grade

In this task is needed to modify the above program so that once the user has completed their chosen action, instead of ending, the program will display the menu again and thus allow the user to select another option. This will require a while loop with indeterminate repetition which is explained in requirement section. Also, should add to menu's option "x – Exit program". If the user enters x (either upper-case or lower-case) then the program should stop looping and end. All new changes are shown at below:

```

#include <stdio.h>
int main(void) {
    char opt='a';
    int num1,num2,i,j;
    float r;
    while(opt!='x')
    {
        printf("\n Please enter the letter which corresponds with your choice");
        printf("\n a – Calculate the area of a rectangle");
        printf("\n b - Calculate the area of a circle");
        printf("\n c – Display a multiplication table");
        printf(" \n d – Add two numbers ");
        printf(" \n X –exit ");
        printf("\n please insert your option: ");
        scanf(" %c", &opt);

        switch (opt)
        {
            .
            .
            .
            case 'x' :
            case 'X' : {

```

```

        printf("end of program! \n");
        opt='x';
        break;
    }
    .
    .
    .
}
getchar();

return 0;
}

```

Initial amount for 'opt' is 'a'. condition for ending while loop is when opt='x'. when the user's choice is X in menu and inserts either 'x' or 'X' then ending message will show and lower-case x goes to opt as ending condition for loop is getting lower-case x and after that the program ends.

```

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
❏ clang-7 -pthread -lm -o main main.c
❏ ./main

Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: a
You have selected a -calcculate the area of a rectangle-----
-----

Please enter the length 2

Please enter the width 3
your result is: 6
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: x
end of program!
-----
❏ 

```

Fig36-taske3-B Grade

A Grade

```
int main(void) {
    char opt;
    int num1,num2,i,j;
    float r;
    do
    {
        printf("Please enter the letter which corresponds with your
        choice");
        printf("\n a - Calculate the area of a rectangle");
        printf("\n b - Calculate the area of a circle");
        printf("\n c - Display a multiplication table");
        printf(" \n d - Add two numbers " );
        printf(" \n X -exit " );
        printf("\n please insert your option: ");
        scanf(" %c", &opt);

        switch (opt)
        {
            case 'A':
            case 'a': {
                printf(" You have selected a -calculate the area of a
                rectangle" );
                printf
                ("-----\n");
                printf("\n Please enter the length ");
                scanf("%d", &num1);
                printf("\n Please enter the width ");
                scanf("%d", &num2);
                printf("your result is: %d ", num1*num2);
                break;
            }
        }
    }
```

Fig37-taske3-A Grade

```
        case 'B':
        case 'b':{
            printf(" You have selected b -Calculate the area of a
            circle");
            printf
            ("-----\n");
            printf("\n Please enter the radius ");
            scanf("%f",&r);
            printf("your result is: %f ", r*r*3.14);
            break;
        }
        case 'C':
        case 'c': {
            printf("You have selected c -Display a multiplication
            table\n");
            printf
            ("-----\n");
            for(i=1;i<=10;i++){
                for(j=1;j<=10;j++){
                    printf("%4d", i*j);
                    printf("\n");
                }
            }
            break;
        }
        case 'D':
        case 'd': {
            printf("You have selected d -add two numbers\n");
```

Fig38-taske3-A Grade

```

case 'D':
case 'd': {
    printf("You have selected d -add two numbers\n");
    printf
    ("-----\n");
    printf("\n Please enter the first number ");
    scanf("%d", &num1);
    printf("\n Please enter the first number ");
    scanf("%d", &num2);
    printf("%d + %d =%4d", num1,num2,num1+num2);
    break;
}
case 'x' :
case 'X' : {
    printf("end of program! \n");
    printf
    ("-----\n");
    opt='x';
    break;
}
default: printf("Sorry, your input was not recognised, please
enter either a, b, c or d");
}

}while(opt!='x');
getchar();
return 0;
}

```

Fig39-taske3-A Grade

As it has been mentioned in requirement section, do..while is another type of loop to making an indeterminate loop. The only different between while and do..while is that while loop checks the condition at beginning of the loop's block but do..while checks the condition at the end of the loop's block. It means do..while runs the loop at least one time even though the condition is false. General statement for do..while is:

Do

{

Statement1

Statement2

.

.

.

}While(ending condition);

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang-7 -pthread -lm -o main main.c
> ./main
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: a
You have selected a -calcculate the area of a rectangle-----
-----

Please enter the length 2

Please enter the width 3
your result is: 6 Please enter the letter which corresponds with your cho
ice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: x
end of program!
-----
> 
```

Fig40-taske3-A Grade

Task 4

D Grade

```
main.c  saved
1  #include <stdio.h>
2
3  int main(void) {
4      int i,j,xcoor,ycoor;
5      char chr;
6      printf("Please enter an X coordinate (a number between 1 and 10): ");
7      scanf("%d",&xcoor);
8      printf("Please enter an y coordinate (a number between 1 and 10): ");
9      scanf("%d",&ycoor);
10     for(i=1;i<=10;i++)
11     {
12         for(j=1;j<=10;j++)
13             if ((xcoor==j)&&(ycoor==i))
14                 printf(" X ");
15             else
16                 printf(" - ");
17             printf("\n");
18         }
19     }
20
21     return 0;
22 }
```

Fig41-taske4-D Grade

For this task, the user should be prompted to enter coordinates. Both coordinates should be represented by a number between 1 and 10. The program should then display a 10x10 square that is made up of dashes. An X should be displayed at the coordinates given by the user. The Entries for this program are xcoor and ycoor which are x and y coordinates. To printing 10*10 square that is made up of dashes is needed nested independent loops. First loop is to counting number of rows and second one is to counting number of columns:

```
for(i=1;i<=10;i++)
    for(j=1;j<=10;j++)
        printf(" - ");
```

here the output is 100 dashes in a row which is wrong output. So is needed to move cursor to next line after printing each row. After printf statement is needed to have printf("\n") before the program runs first loop. If is needed to run printf by first loop, must add brackets as each for without bracket can run just one statement.

```

for(i=1;i<=10;i++)
{
    for(j=1;j<=10;j++)
        printf(" - ");
    printf("\n");
}

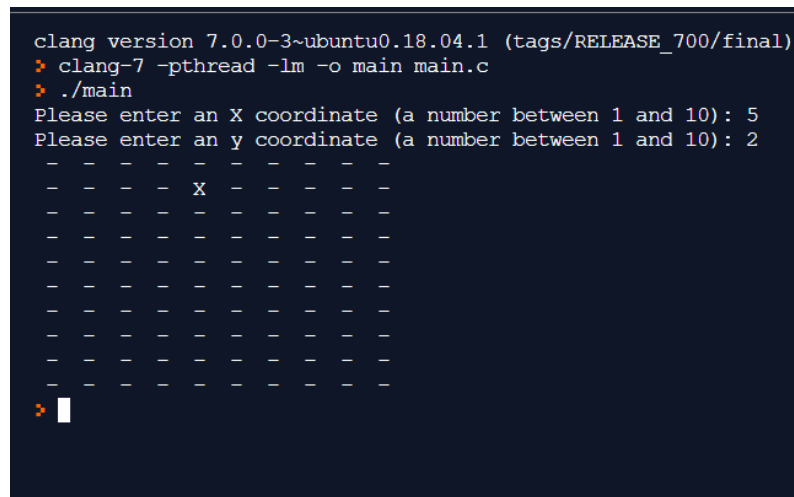
```

So far the output is 10*10 square that is made up of dashes. Consider that the user entered xcoor=5 and ycoor=2. So instead of dash where is in second row and fifth column should be printed a 'X'. it means when xcoor==j and ycoor==i should be printed 'X' instead of dash.

```

for(i=1;i<=10;i++)
{
    for(j=1;j<=10;j++)
        if ((xcoor==j)&&(ycoor==i))
            printf(" X ");
        else
            printf(" - ");
    printf("\n");
}

```



```

clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
Please enter an X coordinate (a number between 1 and 10): 5
Please enter an y coordinate (a number between 1 and 10): 2
- - - - -
- - - X -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
❖ █

```

Fig42-taske4-D Grade

C Grade

```
#include <stdio.h>
#include <stdio.h>
int main(void)
{
    int i,j,xcoor,ycoor;
    char chr,chr2;
    printf("Please enter an X coordinate (a number between 1 and 10): ")
    ;
    scanf("%d",&xcoor);
    printf("Please enter an y coordinate (a number between 1 and 10): ")
    ;
    scanf("%d",&ycoor);
    for(i=1;i<=10;i++)
    {
        for(j=1;j<=10;j++)
            if ((xcoor==j)&&(ycoor==i))
                printf(" X ");
            else
                printf(" - ");
            printf("\n");
    }
}
```

Fig43-taske4-C Grade

```
do
{
    printf("please enter a character of either 'w','a','s', 'd' or 'x' \n 'w': X should have moved up 1 space\n 'a':X should have moved left 1 space\n 's': then the X should have moved down 1 space \n 'd': X should have moved right 1 space \n x: for exit ");

    while ((chr = getchar()) != '\n' && chr != EOF);
    scanf("%c", &chr);

    switch (chr)
    {
        case 'w':
        case 'W':
            {
                ycoor=ycoor-1;
                break;
            }
        case 'a':
        case 'A':
            {
                xcoor=xcoor-1;
                break;
            }
        case 's':
        case 'S':
            {
                ycoor=ycoor+1;
                break;
            }
        ...
    }
}
```

Fig44-taske4-C Grade

```

    case 'd':
    case 'D':
    {
        xcoor=xcoor+1;
        break;
    }
    case 'x' :
    case 'X' :
    {
        break;
    }
    default: printf("Sorry, your input was not recognised, please
    enter either a, b, c or d");
}
    for(i=1;i<=10;i++)
    {
        for(j=1;j<=10;j++)
            if ((xcoor==j)&&(ycoor==i))
                printf(" X ");
            else
                printf(" - ");
            printf("\n");
        }
    }while(chr!='x');
    return 0;
}

```

Fig45-taske4-C Grade

To doing this task is needed to change above program so it now behaves as follows:

- Once the grid has been displayed, the user should be prompted to enter a character of either 'w', 'a', 's', 'd' or 'x'
- The program should then re-print the grid but, depending on what the user entered, the X will now be displayed in a new location.
 - o If the user entered 'w' then the X should have moved up 1 space
 - o If the user entered 'a' then the X should have moved left 1 space
 - o If the user entered 's' then the X should have moved down 1 space
 - o If the user entered 'd' then the X should have moved right 1 space.
 - o If the user entered 'x' then the program should end (see below)
- If the user enters anything other than 'w', 'a', 's', 'd' or 'x' then the program should inform the user of their error

- The program should repeat until the user enters 'x'. However, the user should NOT have to enter the coordinates (from the D grade section) more than once
- Both uppercase and lowercase letters should be accepted.

So is needed to do:

1. Getting xcoor and ycoor from the user.
2. Printing dashes and displaying 'X' at the coordinates given by the user
3. Creating an indeterminate repetition loop which do..while is the best option.
4. Printing a menu which shows to the user 'w','a','s', 'd' or 'x': 'w' for going up, 's' for going down, 'a' for going to the left, 'd' for going to the right
5. Getting user's choice: here there is a problem with getting chars. To solve this problem should be add the following code:

```
while ((chr = getchar()) != '\n' && chr != EOF);
scanf("%c", &chr);
```

chr is user's choice to moving X.

6. Changing either xcoor or ycoor depends on the user's choice through switch..case statement. Depends on 'chr' there are 5 cases:
 - Chr=w then to going up should ycoor=ycoor-1
 - Chr=s then to going down should ycoor=ycoor+1
 - Chr=a then to going left should xcoor=xcoor-1
 - Chr=d then to going right should xcoor=xcoor+1
 - Chr=x then the program should end
7. Reprint the dashes with 'X' at the new coordinates.

As it can be seen when ycoor=2 or 10, X must not move up or down. On the other hand, when xcoor=1 or 10, X must not move left or right. So, when xcoor or ycoor are between 2 to 9 it is possible to move. If the user's choice is going up or down, first should be checked that $2 \leq ycoor \leq 9$. If the condition is false it means that ycoor must not change. . If the user's choice is going left or right, first should be checked that $2 \leq xcoor \leq 9$. If the condition is false it means that xcoor must not change. For this task just needed to change switch case statements.

```
switch (chr)
{
case 'w':
case 'W':
{
    if((ycoor>=2) && (ycoor<=10 ))
        ycoor=ycoor-1;

    else
        ycoor=ycoor;
        break;
}
case 'a':
case 'A':
{
    if ((xcoor>=2)&& (xcoor<=10))
        xcoor=xcoor-1;

    else
        xcoor=xcoor;
        break;
}

case 's':
case 'S':
{
    if ((ycoor<=9) && (ycoor>=2))
        ycoor=ycoor+1;

    else
        ycoor=ycoor;
        break;
}

case 'd':
case 'D':
{
    if ((xcoor<=9) && (xcoor>=2))
        xcoor=xcoor+1;

    else
        xcoor=xcoor;
        break;
}
}
```

Fig49-taske4-B Grade


```

please enter a character of either 'w','a','s', 'd' or 'x'
'w': X should have moved up 1 space
'a':X should have moved left 1 space
's': then the X should have moved down 1 space
'd': X should have moved right 1 space
x: for exit a
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
-  X - - - - -
- - - - -
- - - - -
- - - - -
- - - - -

please enter a character of either 'w','a','s', 'd' or 'x'
'w': X should have moved up 1 space
'a':X should have moved left 1 space
's': then the X should have moved down 1 space
'd': X should have moved right 1 space
x: for exit a
- - - - -
- - - - -
- - - - -
- - - - -
X - - - - -
- - - - -
- - - - -
- - - - -
- - - - -

```

Fig52-taske4-B Grade

```

please enter a character of either 'w','a','s', 'd' or 'x'
'w': X should have moved up 1 space
'a':X should have moved left 1 space
's': then the X should have moved down 1 space
'd': X should have moved right 1 space
x: for exit a
- - - - -
- - - - -
- - - - -
- - - - -
X - - - - -
- - - - -
- - - - -
- - - - -
- - - - -

please enter a character of either 'w','a','s', 'd' or 'x'
'w': X should have moved up 1 space
'a':X should have moved left 1 space
's': then the X should have moved down 1 space
'd': X should have moved right 1 space
x: for exit x
- - - - -
- - - - -
- - - - -
- - - - -
X - - - - -
- - - - -
- - - - -
- - - - -
- - - - -

```

Fig53-taske4-B Grade

A Grade

ASCII code for upper-case letters are 65..90 and for lower-case letters are 97..122. for example, ASCII code for 'a' is 97 and for 'A' is 65. So, if is needed to change an upper-case letter to a lower-case letter just needed to add 32 and for lower-case to upper-case is needed to reduce 32.

Uppercase to lower-case: `chr=chr+32`

Lower-case to upper-case: `chr=chr-32`

Instead of using two continues cases for each letter to accepting both upper-case and lower-case should be used that formula to make sure all characters before starting switch case statement has changed to either upper-case or lower-case. It is decided to change the above program to change all characters to lower-case letter. After getting chr from the user, if it is between 65 and 90 will be added 32 to changing all characters to lower-case.

```
while ((chr = getchar()) != '\n' && chr != EOF);
scanf("%c", &chr);
if (chr>=65 && chr<=90) chr=chr+32;
```

```
while ((chr = getchar()) != '\n' && chr != EOF);
scanf("%c", &chr);
if (chr>=65 && chr<=90) chr=chr+32;
switch (chr)
{
    case 'w':
    {
        if((ycoor>=2) && (ycoor<=10 ))
            ycoor=ycoor-1;

        else
            ycoor=ycoor;
        break;
    }
    case 'a':
    {
        if ((xcoor>=2)&& (xcoor<=10))
            xcoor=xcoor-1;

        else
            xcoor=xcoor;
        break;
    }
    case 's':
    {
```

Fig54-taske4-A Grade

```

    case 's':
    {
        if((ycoor>=1) && (ycoor<=9))
            ycoor=ycoor+1;

        else
            ycoor=ycoor;
        break;
    }

    case 'd':
    {
        if ((xcoor>=1)&& (xcoor<=9))
            xcoor=xcoor+1;

        else
            xcoor=xcoor;
        break;
    }
    case 'x' : {
        break;
    }
    default: printf("Sorry, your input was not recognised, please
enter either a, b, c or d");
}

```

Fig55-taske4-A Grade

```

please enter a character of either 'w','a','s', 'd' or 'x'
'w': X should have moved up 1 space
'a':X should have moved left 1 space
's': then the X should have moved down 1 space
'd': X should have moved right 1 space
x: for exit S
- - - - -
- - - - -
- - - - -
- - - - -
X - - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
please enter a character of either 'w','a','s', 'd' or 'x'
'w': X should have moved up 1 space
'a':X should have moved left 1 space
's': then the X should have moved down 1 space
'd': X should have moved right 1 space
x: for exit S
- - - - -
- - - - -
- - - - -
- - - - -
X - - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

```

Fig56-taske4-A Grade

Task 5

Introduction

Arrays:

Arrays are kind of variable. But the only different between a variable and arrays is that arrays can store multiple values. A variable is a place in RAM which has a name and type. Array is an interconnected space in RAM which can store elements. How can create an array?

Type of array's elements array's name[number of elements]

For example: `Int arr[5];`

That is an array with 5 elements and arr is the name of that array. Each element in array has a number. The numbers are from 0 to 4. It means if an array has N elements then it is numbered from 0 to N-1.

arr	arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
-----	--------	--------	--------	--------	--------

How a programmer can access to an array's element?

Name of the array[the element's number]

For example, `arr[2]`, `arr[1]`.

How a programmer can initialize an array with values? An array can be initialized on creation with values. For example:

`Int arr[]={34,23,2,0,10}`

Here arr is an array with 5 elements:

arr[0]=34	arr[1] =23	arr[2]=2	arr[3]=0	arr[4]=10
-----------	------------	----------	----------	-----------

Also, an array can be initialized after creation.

`Int arr[5];`

`Arr[0]=12;`

`Arr[1]=10;`

```
Arr[2]=13;
```

```
Arr[3]=67;
```

```
Arr[4]=90;
```

It is wiser to initialize an array with a loop as element's numbers are repetitious and can be implemented by loop's counter. If 'N' is the numbers of array's elements then initial amount for counter is 0 and final amount is getting 'N-1'.

For example:

```
Int arr[5];
```

```
For(int counter=0;counter<=4;conter++)
```

```
    Scanf ("%d", &arr[counter]);
```

When counter is equal 0 then the program is waiting to get arr[0] from the user. When counter is equal 1 then the program is waiting to get arr[1] from the user. This process will continue until counter get 5 and loop end. there are two ways to check the final amount for array's loop:

- When counter<=N-1
- When counter<N

Also, it is possible to fill array's elements with while loop :

```
Int arr[5];
```

```
Int counter=0;
```

```
While (counter<=4)
```

```
{
```

```
    Scanf ("%d", &arr[counter]);
```

```
    Counter++;
```

```
}
```

To being more user-friendly when the user is inserting array's elements, is needed a printf statement before scanf statement. it can be including element's number.

```
Printf("please insert element number %d :", &arr[counter+1]);
```

Instead of %d will be written counter+1. For example, when the counter is 0 then user will see 'please insert element number 1'.

For printing array's elements instead of scanf should be written printf statement.

```
Int arr[5]={2,4,7,87,23};  
Int counter=0;  
While (counter<=4)  
{  
    printf("item number %d: %d",counter+1, arr[counter]);  
    printf("\n");  
    counter++;  
}
```

The output is :

Item number 1: 2

Item number 2: 4

Item number 3: 7

Item number 4: 87

Item number 5: 23

D Grade

```
#include <stdio.h>

int main(void) {
    int i, loc, item, sum;
    int arr[6] = {8, 7, 9, 5, 0, 10};
    char chr;
    do
    {
        printf(" \n A - Repopulate array \n B - Display all values \n C -  
Replace one number \n D - Calculate the mean \n x- exit " );
        printf("\n please insert your option: ");

        scanf(" %c", &chr);
        if (chr>=65 && chr<=90) chr=chr+32;

        switch (chr)
        {
            case 'a':
            {
                for(i=0; i<6; i++)
                {
                    printf("please enter your %d item : ", i+1);
                    scanf("%d",&arr[i]);
                }
                break;
            }
            ...
        }
    }
```

Fig57-taske5-D Grade

```
case 'b':
{
    for(i=0; i<6;i++)
    {
        printf("your %d number is:", i);
        printf("%d", arr[i]);
        printf("\n");
    }
    break;
}

case 'c':
{
    printf("please insert your item's location which you want to  
change that: ");
    scanf("%d", &loc);
    printf("please insert your item's number: ");
    scanf("%d", &item);
    arr[loc-1]=item;
    printf(" \n your new item has been replaced!");

    break;
}

case 'd':
{
    for(i=1;i<6;i++)
        sum+=arr[i];
    printf("the avarage is: %d", sum);
    break;
}
..
```

Fig58-taske5-D Grade

```

    case 'd':
    {
        sum=0;
        for(i=0;i<6;i++)
            sum+=arr[i];
        float avg;
        avg= sum/6;
        printf("the avarage is: %f", avg);
        break;
    }
    case 'x' :
    {
        printf("end of program! \n");
        printf("-----\n");
        break;
    }
    default: printf("Sorry, your input was not recognised, please
enter either a, b, c or d");
}

}while(chr!='x');
return 0;
}

```

Fig59-taske5-D Grade

```

A - Repopulate array
B - Display all values
C - Replace one number
D - Calculate the mean
x- exit
please insert your option: a
please enter your 1 item : 1
please enter your 2 item : 3
please enter your 3 item : 43
please enter your 4 item : 45
please enter your 5 item : 65
please enter your 6 item : 32

A - Repopulate array
B - Display all values
C - Replace one number
D - Calculate the mean
x- exit
please insert your option: c
please insert your item's location which you want to change that: 1
please insert your item's number: 12

your new item has been replaced!
A - Repopulate array
B - Display all values
C - Replace one number
D - Calculate the mean
x- exit
please insert your option: b
your 0 number is:12
your 1 number is:3
your 2 number is:43
your 3 number is:45
your 4 number is:65
your 5 number is:32

```

Fig60-taske5-D Grade

```

A - Repopulate array
B - Display all values
C - Replace one number
D - Calculate the mean
x- exit
please insert your option: d
the avarage is: 188
A - Repopulate array
B - Display all values
C - Replace one number
D - Calculate the mean
x- exit
please insert your option: x
end of program!

```

Fig61-taske5-D Grade

C Grade

```

int main(void) {
    int i, loc, item, sum;
    int arr[6] = {8, 7, 9, 5, 0, 10};
    char chr;
    do
    {
        printf("\n A - Repopulate array \n B - Display all values \n C -
        Replace one number \n D - Calculate the mean \n E - finding
        maximun number \n x- exit " );
        printf("\n please insert your option: ");
        scanf(" %c", &chr);
        if (chr>=65 && chr<=90) chr=chr+32;

        switch (chr)
        {
            case 'a':
            {
                for(i=0; i<6; i++)
                {
                    printf("please enter your %d item : ", i+1);
                    scanf("%d",&arr[i]);
                }
                break;
            }
        }
    }
}

```

Fig62-taske5-C Grade


```

    case 'b':
    {
        for(i=0; i<6;i++)
        {
            printf("your %d number is:", i);
            printf("%d", arr[i]);
            printf("\n");
        }
        break;
    }
    case 'c':
    {
        printf("please insert your item's location which you want to
        change that: ");
        scanf("%d", &loc);
        printf("please insert your item's number: ");
        scanf("%d", &item);
        arr[loc-1]=item;
        printf(" \n your new item has been replaced!");
        break;
    }
    case 'd':
    {
        sum=0;
        for(i=0;i<6;i++)
            sum+=arr[i];
        float avg;
        avg= sum/6;
        printf("the avarage is: %f", avg);
        break;
    }
}

```

Fig63-taske5-C Grade

```

    case 'e':
    {
        int max;
        max=arr[0];
        for(i=1;i<6;i++)
            if (arr[i]>max) max=arr[i];
        printf("the maximum number is : %d", max );
        break;
    }
    case 'x' :
    {
        printf("end of program! \n");
        printf("-----\n");
        break;
    }
    default: printf("Sorry, your input was not recognised, please
    enter either a, b, c or d");
}
}while(chr!='x');
return 0;
}

```

Fig64-taske5-C Grade

For this task is needed to Create an array and initialize it with the following numbers: 8, 6, 9, 15, 22 .

```
int arr[6] ={8, 7, 9, 5, 0, 10};
```

to doing this task should be used a 'do..while' loop. In loop's body first should print menu's options. Then getting user's choice. To having less cases in switch

case statement after scanf if chr is an upper-case letter with adding 32 make it a lower-case letter. Final amount to ending loop is getting 'x' character.

```
do
{
    printf("\n A – Repopulate array \n B - Display all values \n C – Replace one number \n D –
Calculate the mean \n E - finding maximum number \n x- exit ");

    scanf(" %c", &chr);
    if (chr>=65 && chr<=90) chr=chr+32;
    .
    .
    .

}while(chr!='x');
```

then should create a menu with following options:

- A – Repopulate array: this option gets array's new elements from the user. As it has been mentioned in requirement section is wiser to use a loop to filling array's elements and it is more user-friendly to use a printf statement before scanf statement. please notice that the user doesn't know that element's number starts from 0 and they just know that the first number is item number 1. So instead of printing I is needed to print i+1 as the user does not know anything about index number in C language.

```
for(i=0; i<6; i++)
{
    printf("please enter your %d item : ", i+1);
    scanf("%d",&arr[i]);
}
```

- B - Display all values: this option should display all the element's values. Same as A menu, is needed a loop but instead of scanf should be used printf statement and after each statement should be moved to next line to printing the next element's value.

```
for(i=0; i<6;i++)
{
    printf("your %d number is:", i+1); //printing item's number
    printf("%d", arr[i]);
    printf("\n"); //moving cursor
}
```

- C – Replace one number: the system should ask the user what element of the array they want to replace. 'loc' is the location where the user wants to replace the item and 'item' is a new value which the user wants to fill arr[loc] with that. Please notice that the user inserts item's number, not index number. For example, if the user inserts 1, it means that 'loc' is equal 0. So, the item should go to arr[loc-1].

```
printf("please insert your item's location which you want to change that: ");
scanf("%d", &loc);
printf("please insert your item's number: ");
scanf("%d", &item);
arr[loc-1]=item;
printf("\n your new item has been replaced!");
```

- D – Calculate the mean: for this option is needed to calculate mean of all the numbers in the array and then divide that to the numbers of elements which is 6. To calculating mean, should add all numbers in a variable. It usually can be named sum. Also is needed to have a loop. Before loop sum=0. In loop's block, should be updated in each loop's repetition with arr[i].

```
sum=0;
for(i=0;i<6;i++)
    sum+=arr[i];
```

So far there is sum of all elements. Now just should be divided to 6 to get the mean. Avg is a float variable to storing the mean.

```
float avg;
avg= sum/6;
printf("the average is: %f", avg);
```

- E - Find largest number: To finding largest number should be assumed the first item in the array is largest number and store that in a variable like 'max' . And then start to compare all elements with max. if second element is larger than max then max should be changed with otherwise max still is keeping first element as largest item. This process continues until all elements compare with max. when comparison finish, max include largest element. It should be implemented by a loop. As max=arr[0] before starting the loop, initial amount for loop's counter should be 1 not 0.

```

int max;
max=arr[0]; //assume first element as largest number before loop
for(i=1;i<6;i++)
    if (arr[i]>max) max=arr[i];
printf("the maximum number is : %d", max );

```

- x- exit: for this option just is needed to print a message and then break statement.

```

A - Repopulate array
B - Display all values
C - Replace one number
D - Calculate the mean
E - finding maximun number
x- exit
please insert your option: a
please enter your 1 item : 23
please enter your 2 item : 45
please enter your 3 item : 65
please enter your 4 item : 234
please enter your 5 item : 9
please enter your 6 item : 0

A - Repopulate array
B - Display all values
C - Replace one number
D - Calculate the mean
E - finding maximun number
x- exit
please insert your option: e
the maximum number is : 234
A - Repopulate array
B - Display all values
C - Replace one number
D - Calculate the mean
E - finding maximun number
x- exit
please insert your option: x
end of program!
-----

```

Fig65-taske5-C Grade

B Grade

```
int main(void) {
    int i, loc, item, sum, max, num;
    int arr[100]={0};
    char chr;
    printf("please insert the number of your items:");
    scanf("%d", &num);
    for(i=0; i<num; i++)
    {
        printf("please enter your %d item : ", i+1);
        scanf("%d",&arr[i]);
    }
    arr[num]=-1;

    do
    {
        printf("\n  A - Repopulate array \n B - Display all values \n C -
        Replace one number \n D - Calculate the mean \n E - finding maximum
        number \n x- exit " );
        printf("\n please insert your option: ");
        // while ((chr = getchar()) != '\n' && chr != EOF);
        scanf(" %c", &chr);
        if (chr>=65 && chr<=90) chr=chr+32;
    }
```

Fig66-taske5-B Grade

```
switch (chr)
{
    case 'a':
    {
        printf("please insert the number of your items:");
        scanf("%d", &num);
        for(i=0; i<num; i++)
        {
            printf("please enter your %d item : ", i+1);
            scanf("%d",&arr[i]);
        }
        arr[num]=-1;
        break;
    }
    case 'b':
    {
        i=0;
        do
        {
            printf("your %d number is:", i);
            printf("%d", arr[i]);
            printf("\n");
            i++;
        }while(arr[i]!=-1);
        break;
    }
}
```

Fig67-taske5-B Grade

```

case 'c':
{
    printf("please insert your item's location which you want to change
that: ");
    scanf("%d", &loc);
    printf("please insert your item's number: ");
    scanf("%d", &item);
    arr[loc-1]=item;
    printf("\n your new item has been replaced!");
    break;
}
case 'd':
{
    i=0;
    do
    {
        sum+=arr[i];
        i++;
    }while(arr[i]!=-1);
    printf("the average is: %d", sum);
    break;
}

```

Fig68-taske5-B Grade

```

case 'e':
{
    max=arr[0];
    i=0;
    do
    {
        if (arr[i]>max) max=arr[i];
        i++;
    }while(arr[i]!=-1);
    printf("the maximum number is : %d", max );
    break;
}
case 'x' :
{
    printf("end of program! \n");
    printf("-----\n");
    break;
}
default: printf("Sorry, your input was not recognised, please enter
either a, b, c or d");
}
}while(chr!='x');
return 0;
}

```

Fig69-taske5-B Grade

To doing this task the above program should behaves as follows:

- Change the array's size to 100: to creating an array with 100 elements :

```
int arr[100]={0};
```

- Before the menu is displayed, the user is forced to enter values into the array. This array has 100 elements. but this is user's choice that how many elements wants to insert into the array. 'num' is a variable which stores taken elements number from the user. Is needed to use a terminator to knowing end of the array

as the user probably will not add 100 elements into the array. . For example, if the user enters 4 then num is 4 and the user should be asked for elements arr[0], arr[1], arr[2] and arr[3]. After getting numbers from the user arr[4] should be filled with -1. So, -1 should be placed into the next free element of the array.

arr[0]=34	arr[1] =23	arr[2]=2	arr[3]=0	arr[4]=-1
-----------	------------	----------	----------	-----------

```
printf("please insert the number of your items:");
scanf("%d", &num);
for(i=0; i<num; i++)
{
    printf("please enter your %d item : ", i+1);
    scanf("%d",&arr[i]);
}
arr[num]=-1;
```

after initializing the array, should be created do..while loop as above program.

- The repopulate array functionality needs to be changed same as first bullet.
- It is needed to change all menu actions to work with terminator. It is not possible to use 'for' loop here because 'for' cannot work with the terminator. So, should be change all 'for' loops to 'do..while' loop. All menu actions should be placed instead of dots. The loop will be end when arr[i] is equal -1.

```
i=0;
do
{
    .
    .
    .
    i++;
}while(arr[i]!=-1);
```

```

please insert the number of your items:4
please enter your 1 item : 1
please enter your 2 item : 2
please enter your 3 item : 6
please enter your 4 item : 76

  A - Repopulate array
  B - Display all values
  C - Replace one number
  D - Calculate the mean
  E - finding maximun number
x- exit
please insert your option: a
please insert the number of your items:3
please enter your 1 item : 2
please enter your 2 item : 1
please enter your 3 item : 5

  A - Repopulate array
  B - Display all values
  C - Replace one number
  D - Calculate the mean
  E - finding maximun number
x- exit
please insert your option: 6
Sorry, your input was not recognised, please enter either a, b, c or
d

```

Fig70-taske5-B Grade

```

  A - Repopulate array
  B - Display all values
  C - Replace one number
  D - Calculate the mean
  E - finding maximun number
x- exit
please insert your option: E
the maximum number is : 5
  A - Repopulate array
  B - Display all values
  C - Replace one number
  D - Calculate the mean
  E - finding maximun number
x- exit
please insert your option: X
end of program!
-----

```

Fig71-taske5-B Grade

A Grade

Getchar():

Instead of using scanf statement to get user's choice for menu, getchar() function can be used. Getchar() is a function which there is in stdio header file. Getchar can read a single character from the user. For example, getting a character from user and then print character's ASCII code. To printing ASCII code is just needed to use %d for chr instead of %c.

```
#include <stdio.h>
int main(void) {
    printf("please insert a character\n");
    char chr;
    chr=getchar();
    printf("%d", chr);
    return 0;
}
```

Finding location of largest number in an array:

In above tasks have shown that how is possible to find largest number in an array. Now finding location of largest number will be show. In this algorithm should be assumed first item in the array is largest number. So, before starting loop max=arr[0] and placemax=0. Just when there is an item which is bigger than max, should be done two jobs:

- Changing max with bigger item: max=arr[counter]
- Change placemax with bigger item's index : placemax=counter;

```
max=arr[0];
placemax=0;
i=0;
do
{
    if (arr[i]>max)
    {
        max=arr[i];
        placemax=i;
    }

    i++;
}while(arr[i]!=-1);
printf("the maximum number is : %d", max );
printf("the location of max number : %d", placemax );
```

Task 6

Introduction

What is a string?

A string is kind of array which stores character elements. since string's length is unpredictable, a terminator which is NULL character use either by system or the programmer. To showing NULL character '\0' should be use.

How can create an array?

Char **name of string**[string length]

For example: **Char** str1[5];

So str1 is a string with 5 characters and '\0' on to the end of the string. System will add NULL to the end of the array automatic.

How a programmer can access to a strings' characters?

Name of string[character's number]

For example, str1[0], str2[1].

How a programmer can initialize a string with characters? A string can be initialized on creation with values. There are many ways to initializing a string:

First way is initializing on creation and add '\0' on to the end of the string by programmer.

chr str1[]={ 'h', 'e', 'l', 'l', 'o', '\0' };

The second way to initializing is initialize on creation but NULL character will add automatically by C.

char str1[]=hello

The third way is getting strings' characters from user. Characters should be got from the user one by one. So is needed to have a loop. Ending condition for loop is getting string's length. The user should be asked to insert each character. After ending loop. The programmer should add NULL character on to the end of the string. For example :

char str1[5];

```

for(i=0;i<5;i++)
{
    printf(" please enter item number %d :", i+1);
    scanf(" %c", &chr);
    str1[i]=chr;
    printf("\n");
}
str1[5]='\0';

```

str1 is a string with 5-character length. On creation C will add '\0' on to the end of str1. The user will be asked each time for a new character and then taken character will add to str. Str1[0] to str1[4] will be filled by the user since ending condition is i<5. After ending loop is needed to add NULL on to the str1[5].

Swapping algorithm values:

How we can swap two values. Consider num1=10 and num2=20. If for swapping values either first num2 move to num1 or first num1 move to num2, one of the values will be missed. To saving both numbers, is needed to have a temporary variable. Then one of the values can be moved to the temporary variable. It can be num1. Then num1 has been saved to the temporary variable. Next step is moving num2 to num1. At the end temporary value will move to num2.

```

Int num1=10;
Int num2=20;
Int temp;
Temp=num1;
Num1=num2;
Num2=temp;

```

D Grade

```
#include <stdio.h>
#include <string.h>
int main(void) {
    int i,choice,count;
    char chr,chrtemp;
    char str1[]={'h','e','l','l','o','\0'};
    char str2[]={'w','r','i','t','e','\0'};
    char strtemp[5];

    do
    {

        printf("\n A - Display a String \n B - Change First String \n C
        - Change Second String " );

        printf("\n please insert your option: ");
        scanf(" %c", &chr);
        if (chr>=65 && chr<=90) chr=chr+32;

        switch (chr)
        {
```

Fig72-taske6-D Grade

```
case 'a':
{
    printf("\n if you would like to print the first string please
    insert 1 and if you would like to print the second string
    please insert 2 (1/2) ? ");

    scanf("%d",&choice);
    if (choice==1)
        strcpy(strtemp, str1);
    else if (choice==2)
        strcpy(strtemp, str2);
    else {
        printf("your entry is wrong");
        break;
    }

    printf("\n");
    int count = 0;
    while(strtemp[count] != '\0')
    {
        printf("%c", strtemp[count]);
        count++;
    }

    printf("\n");
    break;
}
```

Fig73-taske6-D Grade

```

case 'b':
{
    printf("\n");
    for(i=0;i<5;i++)
    {
        printf(" please inter item number %d :", i+1);
        scanf(" %c", &chrtemp);
        str1[i]=chrtemp;
        printf("\n");
    }
    str1[5]='\0';

    break;
}
case 'c':
{
    printf("\n");
    for(i=0;i<5;i++)
    {
        printf(" please inter item number %d :", i+1);
        scanf(" %c", &chrtemp);
        str2[i]=chrtemp;
        printf("\n");
    }
    str1[5]='\0';
    break;
}

default: printf("Sorry, your input was not recognised, please
enter either a, b, c or d");
}

```

Fig74-taske6-D Grade

```

A - Display a String
B - Change First String
C - Change Second String
please insert your option: a

if you would like to print the first string please insert 1 and if you w
ould like to print the second string please insert 2 (1/2) ? 1

hello

A - Display a String
B - Change First String
C - Change Second String
please insert your option: a

if you would like to print the first string please insert 1 and if you w
ould like to print the second string please insert 2 (1/2) ? 2

A - Display a String
B - Change First String
C - Change Second String
please insert your option: b

please inter item number 1 :q
please inter item number 2 :w
please inter item number 3 :e
please inter item number 4 :r
please inter item number 5 :t

```

Fig75-taske6-D Grade

```
A - Display a String
B - Change First String
C - Change Second String
please insert your option: b

please inter item number 1 :q
please inter item number 2 :w
please inter item number 3 :e
please inter item number 4 :r
please inter item number 5 :t

A - Display a String
B - Change First String
C - Change Second String
please insert your option: a

if you would like to print the first string please insert 1 and if you
ould like to print the second string please insert 2 (1/2) ? 1

qwert

A - Display a String
B - Change First String
C - Change Second String
please insert your option: 
```

Fig76-taske6-D Grade

C Grade

```
#include <stdio.h>
#include <string.h>
int main(void) {
    int i,choice,count;
    char chr,chrtemp;
    char str1[]={'h','e','l','l','o','\0'};
    char str2[]={'w','r','i','t','e','\0'};
    char strtemp[5];

    do
    {

        printf("\n A - Display a String \n B - Change First String \n C
        - Change Second String \n D - swap strings \n X -exit " );

        printf("\n please insert your option: ");
        scanf(" %c" , &chr);
        if (chr>=65 && chr<=90) chr=chr+32;

        switch (chr)
        {
```

Fig77-taske6-C Grade

```
        case 'd':
        {
            printf("\n");

            for(i=0;i<5;i++)
            {
                chrtemp=str1[i];
                str1[i]=str2[i];
                str2[i]=chrtemp;
            }
            printf("strings has been changed!!");
            printf("\n");

            break;
        }
        case 'x':
        {
            printf("end of the program");
            break;
        }

        default: printf("Sorry, your input was not recognised, please
        enter either a, b, c or d");
    }

    }while(chr!='x');
    return 0;
}
```

Fig78-taske6-C Grade

For this task is needed to create two strings with same length and give them value. Two strings should be ended by '\0'.

```
char str1[]={'h','e','l','l','o','\0'};
char str2[]={'w','r','i','t','e','\0'};
```

should be created a menu with following options:

- Display a String: since there are two strings, the user should be asked that which strings wants to display. Strcpy is a C function in string header file which will copy the second string into the first string.

Strcpy(the first string, the second string)

So, the second string will move to the first string. Strtemp is a string which will be printed depends on the user's choice. If the user's choice is 1 then it means they want to see str1, so str1 will be move to the strtemp. If the user's choice is 2 then it means they want to see str2, so str2 will be move to the strtemp. If the users insert another input, it means taken input is wrong and after showing a message, break statement will be run.

```
printf("\n if you would like to print the first string please insert 1 and if you would like to print
the second string please insert 2 (1/2) ? ");
char strtemp[5];
scanf("%d",&choice);
if (choice==1)
    strcpy(strtemp, str1);
else if (choice==2)
    strcpy(strtemp, str2);
else {
    printf("your entry is wrong");
    break;
}
```

So far user's chosen string is into the strtemp. Should be have a loop with a counter which ending condition is getting a NULL character.

```
count = 0;

while(strtemp[count] != '\0')
{
    printf("%c", strtemp[count]);
    count++;
}
```


strings' characters will be print character by character. When count is 0 then strtemp[0] will be print and after count++, strtemp[1] will be print. All characters will be print until getting NULL character. NULL character means end of the string.

- Change First String : To changing the first string is needed to get 5 characters from the user. Having another string is not necessary as it can be done by having extra character variable. Should be done by a loop with 5 repetition from 0 to 4. It can be $i < 5$ or $i \leq 4$. Just must get the user's which is chrtemp and then move that to its location in str1. New character's location can be found by loop's counter which is 'i'.

```
char chrtemp;
for(i=0;i<5;i++)
{
    printf(" please enter item number %d :", i+1);
    scanf(" %c", &chrtemp);
    str1[i]=chrtemp;
    printf("\n");
}
str1[5]='\0';
```

after loop must be added NULL character onto the str1[5].

- Change Second String: This option is same as above option. Just instead of str1[i] should be written str2[i].
- Swapping two strings: To swapping two strings is needed to define a character variable. Should be used swapping values algorithm which is explained in requirement section. Is needed a loop. Loop's repetition depends on strings' length which here is 5. In addition, it can be done by while loop with getting NULL character to ending loop. Just need to read the first character from the first string and then store that into chrtemp and then read first character from second string and then move that into the first location of the first string and . Finally, should be moved chrtemp into the first location of second string. This process should be done 5 times for all characters.

```

for(i=0;i<5;i++)
{
    chrtemp=str1[i];
    str1[i]=str2[i];
    str2[i]=chrtemp;
}
printf("strings has been changed!!!");

```

it is more user-friendly when after finishing copy process, print a message for the user.

- Exit. Just is needed to add a break statement.

```

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
X -exit
please insert your option: d

strings has been changed!!

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
X -exit
please insert your option: a

if you would like to print the first string please insert 1 and if you w
ould like to print the second string please insert 2 (1/2) ? 2

hello

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
X -exit
please insert your option: a

if you would like to print the first string please insert 1 and if you w
ould like to print the second string please insert 2 (1/2) ? 1

write

```

Fig79-taske6-C Grade

```

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
X -exit
please insert your option: a

if you would like to print the first string please insert 1 and if you
ould like to print the second string please insert 2 (1/2) ? 1

write

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
X -exit
please insert your option: x
end of the program

```

Fig80-taske6-C Grade

B Grade

```
case 'e':
{
    printf("\n");
    int equal=0;
    count = 0;
    while(str1[count] != '\0' && equal!=1)
    {
        if (str1[count]!=str2[count]) equal=1;
        count++;
    }
    if (equal==1) printf("the two strings are different!");
    else printf("the two strings are identical");
    printf("\n");
    break;
}
case 'x':
{
    printf("end of the program");
    break;
}
```

Fig81-taske6-B Grade

For this task should be added to the menu a new option which is comparing two strings. If they are the same, then "The two strings are identical" should be displayed. Otherwise It should display "The two strings are different". To implementing this task must be used a loop. also, is needed a variable same as a flag. In this task the flag's variable is named 'equal' with initial amount of zero. As soon as finding different characters in the same position in both strings, 'equal' should be changed to one. Ending condition for loop is a collection of two conditions. One of them is when 'equal' is 1 and the other one is getting NULL character. Since length of two strings is same, each one of them can be used. Until equal=0 and str1[count]!=null loop continues to work. When loop finishes and still is equal zero means both strings are same otherwise they are different.

```
int equal=0;
count = 0;
while(str1[count] != '\0' && equal!=1)
{
    if (str1[count]!=str2[count]) equal=1;
    count++;
}
if (equal==1) printf("the two strings are different!");
else printf("the two strings are identical");
printf("\n");
```

```

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
E - Comparing two atrings
X -exit
please insert your option: e

the two atrings are different!

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
E - Comparing two atrings
X -exit
please insert your option: b

please inter item number 1 :h
please inter item number 2 :e
please inter item number 3 :l
please inter item number 4 :l
please inter item number 5 :o

```

Fig82-taske6-B Grade

```

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
E - Comparing two atrings
X -exit
please insert your option: b

please inter item number 1 :h
please inter item number 2 :e
please inter item number 3 :l
please inter item number 4 :l
please inter item number 5 :o

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
E - Comparing two atrings
X -exit
please insert your option: e

the two atrings are different!

A - Display a String
B - Change First String
C - Change Second String
D - swap strings
E - Comparing two atrings
X -exit
please insert your option: x
end of the program

```

Fig83-taske6-B Grade

A Grade

String.h library has many pre-written functions which can be used easily, and it is simple to work on strings with them. To using these pre-written functions must introduce String.h library at the beginning of the program before main().

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

Here are some of them:

- sizeof(): with this function can get string's length. For example:

```
char str1[]="hello";
```

```
printf("%d", sizeof(str1));
```

the output is 6 because '\0' is into the end of string.

This function can be used for ending conditions in loops:

```
For(i=0;i<sizeof(str1); i++)
```

Also, instead of `i<sizeof(str1)` can be used `i<=sizeof(str1)-1`.

- Strcpy(): this function will copy second string into first string.

```
char s1[]="hello";
```

```
char s2[]="abcde";
```

```
strcpy(s2,s1);
```

```
printf("%s", s2);
```

The output is 'hello' because s1 copied into the s2.

- Strcmp(): this function compares two strings and returns a number
There are 3 type of output for that:
 - ✓ If two strings are the same, the returned number is equal zero
 - ✓ If the first string is bigger than second one then the returned number is a positive number.
 - ✓ If the first string is less than second one then the returned number is a negative number.

This function compares two strings depend on their ASCII codes.

```
main.c saved
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void) {
5     char s1[]="hello";
5     char s2[]="abcde";
7     printf(" \n %d", strcmp(s1,s2));
3
9     char s3[]="abcde";
9     char s4[]="hello";
1    printf(" \n %d", strcmp(s3,s4));
2
3    char s5[]="hello";
4    char s6[]="hello";
5    printf(" \n %d", strcmp(s5,s6));
5    return 0;
7 }
```

```
https://te
clang ve
clang-
./main
7
-7
0
```

Fig84-taske6-A Grade

In fig67 there are three examples for work with Strcmp(). In first two strings, s1 is bigger than s2, so the output is 7 which is a positive number. In second example, s4 is less than s3, thus the output is a negative number. In last example, the output is zero since both strings are same.

- Strcat(): with this function can be added a string at the end of another string. The result will insert into the first string.

```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(void) {
5     char s1[]="hello";
6     char s2[]="friends";
7     char s3[]="";
8     strcat(s1, s2);
9     printf("%s", s1);
10
11
12     return 0;
13 }
```

```
clang version 7.0.0-3~u
(tags/RELEASE_700/final
clang-7 -pthread -lm
./main
hellofriends
```

Fig85-taske6-A Grade

In fig85 s2 will insert at to the end of s1. The result is in s1.

Task 7

D Grade

```
#include <stdio.h>

int main(void)
{
    int counter,i,j,temp;
    int arr[5]={6,5,3,1,2};
    for(i=0;i<=4;i++)
    {
        printf(" arr[0]=%d  arr[%d]=%d", arr[0],i,
        arr[i]);
        printf("\n");
    }

    return 0;
}
```

```
> clang-7 -pthread -lm -o :
> ./main
arr[0]=6  arr[0]=6
arr[0]=6  arr[1]=5
arr[0]=6  arr[2]=3
arr[0]=6  arr[3]=1
arr[0]=6  arr[4]=2
> □
```

Fig86-taske7-D Grade

C Grade

Implementation of a Sorting Algorithm step by step: A sorting algorithm can sort elements of an array either Ascending or descending.

For this task should be done following steps:

- Should be created an array with following numbers: 6 5 3 1 2

```
int arr[5]={6,5,3,1,2};
```

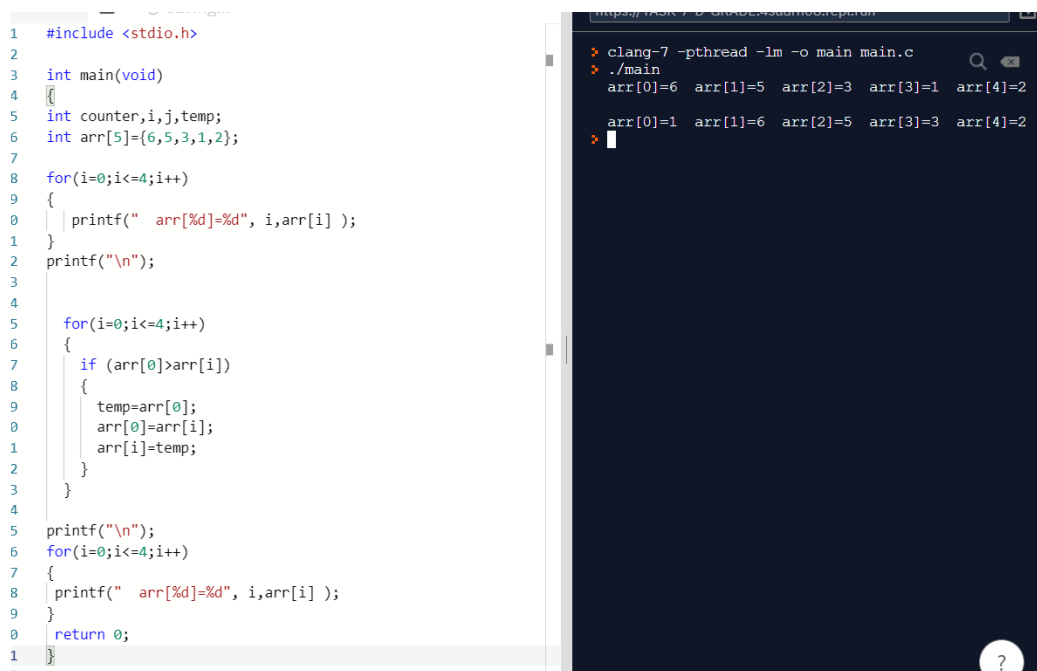
- Implement a loop that iterates through the array and For each iteration of the loop need to checks to see if the current element that the loop is accessing is smaller than the value stored in element 0 – if it is, the two values should be swapped over. Initial amount for loop is 0 and ending condition is getting 4. Should be compared for each iteration should be compared arr[0] with current element that the loop is accessing. the current element is arr[i]. if arr[i] is smaller than arr[[0] then they should be swapped. The swapping algorithm is same as the one which has been used in TASK 6. To swapping is needed to have a temp variable otherwise

one of the values will be missed. After this loop the lowest number will be at the beginning of the array.

```
for(i=0;i<=4;i++)
{
    if (arr[0]>arr[i])
    {
        temp=arr[0];
        arr[0]=arr[i];
        arr[i]=temp;
    }
}
```

- Print the array out before the loop and afterwards to make sure that it gives the correct result. To printing array's value just need a loop from 0 to 4 .

```
for(i=0;i<=4;i++)
printf(" arr[%d]=%d", i,arr[i] );
```



The image shows a C program in a code editor on the left and its execution output in a terminal on the right. The program defines an array of 5 integers and uses a selection sort algorithm to sort it in ascending order. It prints the array before and after the sorting process.

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     int counter,i,j,temp;
6     int arr[5]={6,5,3,1,2};
7
8     for(i=0;i<=4;i++)
9     {
10         printf(" arr[%d]=%d", i,arr[i] );
11     }
12     printf("\n");
13
14     for(i=0;i<=4;i++)
15     {
16         if (arr[0]>arr[i])
17         {
18             temp=arr[0];
19             arr[0]=arr[i];
20             arr[i]=temp;
21         }
22     }
23
24     printf("\n");
25     for(i=0;i<=4;i++)
26     {
27         printf(" arr[%d]=%d", i,arr[i] );
28     }
29     return 0;
30 }
```

The terminal output shows the array before sorting: arr[0]=6 arr[1]=5 arr[2]=3 arr[3]=1 arr[4]=2. After sorting, the output is: arr[0]=1 arr[1]=6 arr[2]=5 arr[3]=3 arr[4]=2.

Fig87-taske7-C Grade

B Grade

To sorting the rest of the array, is needed another loop. For this type of sorting if an array has N elements then is needed N-1 steps for new loop. When j=0 the algorithm will compare element 0 to all elements below it. It means the second loop should be have initial amount of 'j+1'. Now the smallest number moves to the top of array. When j gets 1, second loop works on elements 2 to 4. If there is an item which is smaller than element 1 then they will be swap. should follow this pattern until it reaches the end of the array.

```
for(j=0;j<4;j++)
{
    for(i=j+1;i<=4;i++)
    {
        if (arr[j]>arr[i])
        {
            temp=arr[j];
            arr[j]=arr[i];
            arr[i]=temp;
        }
    }
}
```

```
2
3 int main(void)
4 {
5     int counter,i,j,temp;
6     int arr[5]={6,5,3,1,2};
7
8     for(i=0;i<=4;i++)
9     {
10         printf(" arr[%d]=%d", i,arr[i] );
11     }
12     printf("\n");
13
14     for(j=0;j<4;j++)
15     {
16         for(i=j+1;i<=4;i++)
17         {
18             if (arr[j]>arr[i])
19             {
20                 temp=arr[j];
21                 arr[j]=arr[i];
22                 arr[i]=temp;
23             }
24         }
25     }
26
27     printf("\n");
28     for(i=0;i<=4;i++)
29     {
30         printf(" arr[%d]=%d", i,arr[i] );
31     }
32     return 0;
33 }
```

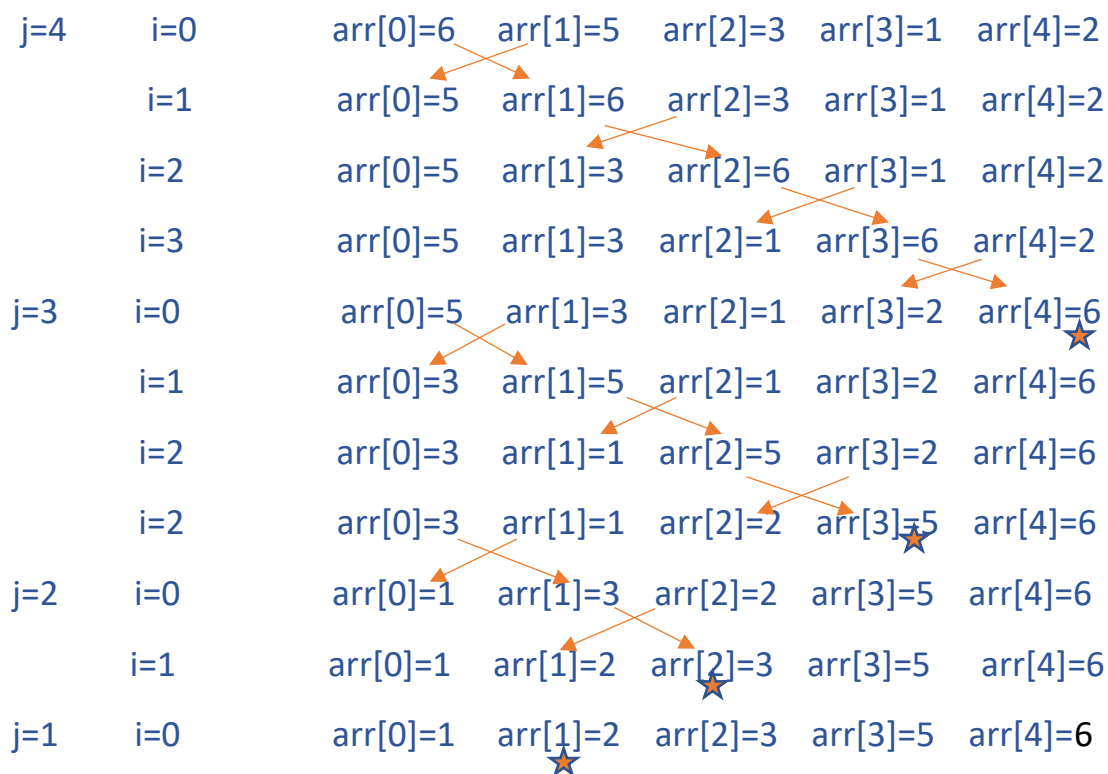
```
clang-7 -pthread -lm -o main main.c
./main
arr[0]=6 arr[1]=5 arr[2]=3 arr[3]=1 arr[4]=2
arr[0]=1 arr[1]=2 arr[2]=3 arr[3]=5 arr[4]=6
```

Fig88-taske7-B Grade

A Grade

Implementation of bubble sort:

For this sort is needed nested loop. First loop is descending loop and second loop is ascending loop. Consider an array with 5 elements. first loop should be having 4 descending iteration from 4 to 1. And the second loop should be having ascending iteration same as current number of first loop. It means if there are N elements in the array then first loop is from N to 1 and the second loop is from 0 to counter amount in the first loop. At the end of each iteration of the first loop, biggest element will move to $arr[j]$. second loop compares pair of elements each time and if the second one is smaller than the first one, will swap them.



```

1  #include <stdio.h>
2  int main(void)
3  {
4  int counter,i,j,temp;
5  int arr[5]={6,5,3,1,2};
6
7  for(i=0;i<=4;i++)
8  {
9      printf("  arr[%d]=%d", i,arr[i] );
10 }
11 printf("\n");
12
13 for(j=4;j>0;j--)
14 {
15     for(i=0;i<j;i++)
16     {
17         if(arr[i] > arr[i+1])
18         {
19             temp = arr[i];
20             arr[i] = arr[i+1];
21             arr[i+1] = temp;
22         }
23     }
24 }
25 printf("\n");
26 for(i=0;i<=4;i++)
27 {
28     printf("  arr[%d]=%d", i,arr[i] );
29 }
30 return 0;
31 }

```

```

❏ clang-7 -pthread -lm -o main main.c
❏ ./main
arr[0]=6  arr[1]=5  arr[2]=3  arr[3]=1  arr[4]=2

arr[0]=1  arr[1]=2  arr[2]=3  arr[3]=5  arr[4]=6

```

Fig89-taske7-A Grade

Task 8

Introduction

Requirement to doing this Task:

What is a function? A function is a section of a code which place in a separate section of code and can execute a task. Each function can either return an amount to main body of program or can do everything into the its body. Let us to talk about functions in real. Consider someone ask you to do something:

- Case 1: It can be for example washing your hand, so you just need to go and wash your hand. In this case you did not get something from that person, and you did not give back something to the person as well.

- Case 2: it can be for example asking to borrow a book from you. Then you just need to go and bring the book for the person. So, in this case, the person is waiting to get the book from you.
- Case 3: it can be do shopping for the person. Then you need to get the money from the person and then take all shopping and the change for the person.

So, to writing a function just is needed determine what a function will be do. A function will be called by Main function, but the body of function is out of Main(). General structure of a function:

Type of function's output name of function(function's requirement)

```
{
    Body of function
}
```

what is type of function's output? A function can have either one output or nothing. If a function does not have any output, then type of function's output is void. Type of function can be integer, char or anything else. If type of function is not void, then at the end of function must be used return statement. Return statement can be a value or variable. If function to finishing its job is needed to get something from main function. Let us explain all three cases with an example. Design three functions, F1 gets two numbers from the user and prints them, all the actions should be done in the body of function. F2 should be get all inputs from the main function. F3 gets its requirements from the main function and just calculates the result, but the result should be print in the main function.

```
#include <stdio.h>
/////a place for introducing functions headers
void f1();
void f2(int a,int b);
int f3(int a,int b);

int main(void) {
    int a,b,r;
    f1(); //recalling f1 by the main function
    /////
    scanf("%d%d", &a,&b);
    f2(a,b); // recalling f2 by the main and sending its requirement
    /////
    scanf("%d%d", &a,&b);
    r=f3(a,b); // recalling f3 by the main function and send its requirements and put the result into a
    variable
```

```

printf("%d",r);
return 0;
}
void f1()
{
    int a,b;
    scanf("%d%d", &a,&b);
    printf("%d", a+b);
}
void f2(int a,int b)
{
    printf("%d", a+b);
}
int f3(int a,int b)
{
    int s;
    s=a+b;
    return s;
}

```

The header of all functions should be introduced at the beginning of the program before main(). F1 does not have any requirement from the main program. So, just is needed to recall the function with f1() statement and in the header, function's requirement section is empty. F1 will not return any output to the main function therefore Type of function's output is void. Into f1 is taken inputs an calculated and printing the result. F2 needs to get a and b from the main program. Therefore, in the main function a and b should be sent to f2 function. function's requirements in f2 are introducing two variables which the main program will send to the function. F2 is needed to calculate and print the output. finally, f3 must does only calculating the output and then return the result to the main program. So, type of output is integer as it wants to send the result to the main. Into body of f3 just is needed to calculate the result and then return that to the main program. In the main program function's result will go to a variable. To printing the result should be used the variable which here is 'r'.

D Grade

```
#include <stdio.h>
void area_rectangle();
void area_circle();
void multiplication_table();
void add_two_numbers();

char opt;
int num1,num2,i,j;
float r;

int main(void) {

    do
    {
        printf("Please enter the letter which corresponds with your choice");
        printf("\n a - Calculate the area of a rectangle");
        printf("\n b - Calculate the area of a circle");
        printf("\n c - Display a multiplication table");
        printf(" \n d - Add two numbers ");
        printf(" \n X -exit ");
        printf("\n please insert your option: ");
        scanf(" %c", &opt);
        if (opt>=65 && opt<=90) opt=opt+32;
```

Fig90-taske8-D Grade

```
switch (opt)
{
case 'a': {
    printf(" You have selected a -calcculate the area of a rectangle" );
    area_rectangle();
    break;
}
case 'b':{
    printf(" You have selected b -Calculate the area of a circle");
    area_circle();
    break;
}
case 'c': {
    printf("You have selected c -Display a multiplication table\n");
    multiplication_table();
    break;
}
case 'd': {
    printf("You have selected d -add two numbers\n");
    add_two_numbers();
    break;
}
case 'x' : {
    printf("end of program! \n");
    printf("-----\n");
    break;
}
default: printf("Sorry, your input was not recognised, please enter either a,
b, c or d");
}
```

Fig91-taske8-D Grade

```

    }while(opt!='x');
    getchar();
    return 0;
}
void area_rectangle()
{
    printf("\n Please enter the length ");
    scanf("%d" , &num1);
    printf("\n Please enter the width ");
    scanf("%d" , &num2);
    printf("your result is: %d ", num1*num2);
}
void area_circle()
{
    printf("\n");
    printf("\n Please enter the radius ");
    scanf("%f",&r);
    printf("your result is: %f ", r*r*3.14);
}
void multiplication_table()
{
    printf("\n");
    for(i=1;i<=10;i++){
        for(j=1;j<=10;j++){
            printf("%4d", i*j);
            printf("\n");
        }
    }
}

```

Fig92-taske8-D Grade

```

void multiplication_table()
{
    printf("\n");
    for(i=1;i<=10;i++){
        for(j=1;j<=10;j++){
            printf("%4d", i*j);
            printf("\n");
        }
    }
}
void add_two_numbers()
{
    printf("\n");
    printf("\n Please enter the first number ");
    scanf("%d" , &num1);
    printf("\n Please enter the first number ");
    scanf("%d" , &num2);
    printf("%d + %d =%4d", num1,num2,num1+num2);
}

```

Fig93-taske8-D Grade

```

./main
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: a
You have selected a -calcculate the area of a rectangle
Please enter the length 1

Please enter the width 2
your result is: 2 Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: b
You have selected b -Calculate the area of a circle

Please enter the radius 1
your result is: 3.140000 Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: c
❖ clang-7 -pthread -lm -o main main.c
❖ clang-7 -pthread -lm -o main main.c
❖ ./main

```

Fig94-taske8-D Grade

```

Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: c
❖ clang-7 -pthread -lm -o main main.c
❖ clang-7 -pthread -lm -o main main.c
❖ ./main

Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: a
You have selected a -calcculate the area of a rectangle
Please enter the length 2

Please enter the width 4
your result is: 8
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: b
You have selected b -Calculate the area of a circle

Please enter the radius 3
your result is: 28.260000

```

Fig95-taske8-D Grade


```

1  2  3  4  5  6  7  8  9 10
2  4  6  8 10 12 14 16 18 20
3  6  9 12 15 18 21 24 27 30
4  8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
6 12 18 24 30 36 42 48 54 60
7 14 21 28 35 42 49 56 63 70
8 16 24 32 40 48 56 64 72 80
9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100

Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: d
You have selected d -add two numbers

Please enter the first number x

Please enter the first number 2 + 4 = 6
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: end of program!
-----

```

Fig96-taske8-D Grade

C Grade

```

#include <stdio.h>
void area_rectangle(int,int);
void area_circle(float);
void multiplication_table();
void add_two_numbers(int,int);

char opt;
int out,num1,num2,i,j;
float r;

int main(void)
{
    do
    {
        printf("\n");
        printf("Please enter the letter which corresponds with your choice");
        printf("\n a - Calculate the area of a rectangle");
        printf("\n b - Calculate the area of a circle");
        printf("\n c - Display a multiplication table");
        printf("\n d - Add two numbers ");
        printf("\n X -exit ");
        printf("\n please insert your option: ");
        scanf(" %c", &opt);
        if (opt>=65 && opt<=90) opt=opt+32;
    }
}

```

Fig97-taske8-C Grade

```

switch (opt)
{
case 'a': {
printf(" You have selected a -calcculate the area of a
rectangle" );
printf("\n Please enter the length ");
scanf("%d" , &num1);
printf("\n Please enter the width ");
scanf("%d" , &num2);
area_rectangle(num1,num2);
break;
}
case 'b':{
printf(" You have selected b -Calculate the area of a
circle");
printf("\n");
printf("\n Please enter the radius ");
scanf("%f",&r);
area_circle(r);

break;
}
case 'c': {
printf("You have selected c -Display a multiplication
table\n");
multiplication_table();
break;
}
}

```

Fig98-taske8-C Grade

```

case 'c': {
printf("You have selected c -Display a multiplication
table\n");
multiplication_table();
break;
}
case 'd': {
printf("You have selected d -add two numbers\n");
printf("\n");
printf("\n Please enter the first number ");
scanf("%d" , &num1);
printf("\n Please enter the first number ");
scanf("%d" , &num2);
add_two_numbers(num1,num2);
break;
}
case 'x' : {
printf("end of program! \n");
printf("-----\n");
;
break;
}
default: printf("Sorry, your input was not recognised, please enter
either a, b, c or d");
}

}while(opt!='x');
getchar();
return 0;
}

```

Fig99-taske8-C Grade

```

        printf("-----\n"
        ;
        break;
    }
    default: printf("Sorry, your input was not recognised, please enter
    either a, b, c or d");
}

}while(opt!='x');
getchar();
return 0;
}
void area_rectangle(int num1,int num2)
{
    printf("your result is : %d",num1*num2);
}
void area_circle(float r)
{
    printf("your result is: %f ", r*r*3.14);
}
void multiplication_table()
{
    printf("\n");
    for(i=1;i<=10;i++){
        for(j=1;j<=10;j++)
            printf("%4d", i*j);
        printf("\n");
    }
}

```

Fig100-taske8-C Grade

```

void area_rectangle(int num1,int num2)
{
    printf("your result is : %d",num1*num2);
}
void area_circle(float r)
{
    printf("your result is: %f ", r*r*3.14);
}
void multiplication_table()
{
    printf("\n");
    for(i=1;i<=10;i++){
        for(j=1;j<=10;j++)
            printf("%4d", i*j);
        printf("\n");
    }
}
void add_two_numbers(int num1, int num2)
{
    printf("your result is : %d", num1+num2);
}

```

Fig101-taske8-C Grade

For this task, should be changed the code that has been Witten for Assessment Sheet 3 and then Modify your code from Assessment Sheet 3 so that instead of doing all jobs into switch case's body, should be recalled a function. For example, if the user enters 'A', then the program should recall a function which calculates the area of rectangle. The functions should not contain any scanf statements.

To calculating the area of the rectangle to getting num1 and num2 is needed to put scanf statements in the switch case body and then send that to the area-rectangle function by `area_rectangle(num1,num2)` statement. header of the function is `void area_rectangle(int num1,int num2)` as it will not return any item to the main function so type of output is void and its requirements which will be sent by the main program are `int num1,int num2`. The area-rectangle function will do calculating and printing the output into its body.

For calculating the area of circle should be got inputs in the main body and then sent them to the function by `area_circle(r)` statement. 'r' is circle's radius which is a float variable. `area_circle` function does not return any result to the main function and it requires to get 'r' from main function. So, the Header of the function is `void area_circle(float r)` as it does not return. In the body of function just is needed to have a printf statement. `multiplication_table` function does not need any requirement from the main function and does not return any result, so the header of the function is `void multiplication_table()`. Finally `add_two_numbers` function does not have any output and gets its requirement from the main function. So, the header of that is `void add_two_numbers(int num1, int num2)`. As all the functions have been defined after the main program, they should be introduced to the system at the beginning of the program. Otherwise the system will not know them. So, they should be placed before main:

```
void area_rectangle(int,int);  
void area_circle(float);  
void multiplication_table();  
void add_two_numbers(int,int);
```

for example, when the system reads first header knows there is a function after main function which does not return any result to the main function and the function needs to get two integer variables from the main function to finishing its process.

```

> clang-7 -pthread -lm -o main main.c
> ./main

Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: b
You have selected b -Calculate the area of a circle

Please enter the radius 2
your result is: 12.560000
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option:
x
end of program!
-----
> █

```

Fig102-taske8-C Grade

B Grade

```

#include <stdio.h>
int area_rectangle(int,int);
float area_circle(float);
void multiplication_table();
int add_two_numbers(int,int);
int multiplyTwoNumbers(int,int);

char opt;
int out,num1,num2,i,j;
float r;

int main(void)
{
    do
    {
        printf("\n");
        printf("Please enter the letter which corresponds with your choice");
        printf("\n a - Calculate the area of a rectangle");
        printf("\n b - Calculate the area of a circle");
        printf("\n c - Display a multiplication table");
        printf(" \n d - Add two numbers ");
        printf(" \n X -exit ");
        printf("\n please insert your option: ");
        scanf(" %c", &opt);
        if (opt>=65 && opt<=90) opt=opt+32;

        switch (opt)

```

Fig103-taske8-B Grade

```

switch (opt)
{
case 'a': {
    int result;
    printf(" You have selected a -calculate the area of a rectangle" );
    printf("\n Please enter the length ");
    scanf("%d" , &num1);
    printf("\n Please enter the width ");
    scanf("%d" , &num2);
    result= area_rectangle(num1,num2);
    printf("your result is: %d ", result);
    break;
}
case 'b':{
    float result;
    printf(" You have selected b -Calculate the area of a circle");
    printf("\n");
    printf("\n Please enter the radius ");
    scanf("%f",&r);
    result = area_circle(r);
    printf("your result is: %f", result);
    break;
}
case 'c': {
    printf("You have selected c -Display a multiplication table\n");
    multiplication_table();
    break;
}
}

```

Fig104-taske8-B Grade

```

    case 'd': {
        int result;
        printf("You have selected d -add two numbers\n");
        printf("\n");
        printf("\n Please enter the first number ");
        scanf("%d" , &num1);
        printf("\n Please enter the first number ");
        scanf("%d" , &num2);
        result= add_two_numbers(num1,num2);
        printf("your result is: %d", result);
        break;
    }
    case 'x' : {
        printf("end of program! \n");
        printf("-----\n");
        break;
    }
    default: printf("Sorry, your input was not recognised, please enter either a, b, c or d");
}

}while(opt!='x');
getchar();
return 0;
}
int area_rectangle(int num1,int num2)
{
    return(num1*num2);
}

```

Fig105-taske8-B Grade

```

float area_circle(float r)
{
    return(r*r*3.14);
}
void multiplication_table()
{
    printf("\n");
    for(i=1;i<=10;i++){
        for(j=1;j<=10;j++){
            printf("%4d", multiplyTwoNumbers(i,j) );
        }
        printf("\n");
    }
}
int multiplyTwoNumbers(int a, int b)
{
    return a*b;
}
int add_two_numbers(int num1, int num2)
{
    return(num1+num2);
}

```

Fig106-taske8-B Grade

For this task, all scanf and printf statements should be done in the main program. For area_rectangle function num1 and num2 gets from the user in switch-case body. The function will be recalled by area_rectangle(num1,num2);

Statement. To saving the result which will be send to the main program by the function, it is been used 'result' variable. The variable which is used to saving the result and function's output must be same type. In this case 'result' and function's output's type are integer. For area_circle function type of 'result' variable and functions' output should be both float. For add_two_numbers the 'result' variable should be integer as function's output is integer.

Into all functions should be used return statement to sending function's output to the main program. For example, return(num1*num2). If type of function's output is not void then the function must use return statement to be sending its output to the main program.

Sometimes is needed to recall a function by another function. In this task should be changed the multiplication_table function so that to calculating multiplication of i and j needs to recall multiplyTwoNumbers function and send i and j as function's requirements. In each loop's iteration printf("%4d",

multiplyTwoNumbers(i,j)); will be run. 'i' and 'j' are multiplyTwoNumbers function requirements. As type of the multiplyTwoNumbers output is integer then should be used %d in printf statement.

```
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: c
You have selected c -Display a multiplication table

 1  2  3  4  5  6  7  8  9 10
 2  4  6  8 10 12 14 16 18 20
 3  6  9 12 15 18 21 24 27 30
 4  8 12 16 20 24 28 32 36 40
 5 10 15 20 25 30 35 40 45 50
 6 12 18 24 30 36 42 48 54 60
 7 14 21 28 35 42 49 56 63 70
 8 16 24 32 40 48 56 64 72 80
 9 18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100

Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: a
You have selected a -calcculate the area of a rectangle
Please enter the length 2

Please enter the width 3
your result is: 6
Please enter the letter which corresponds with your choice
a - Calculate the area of a rectangle
b - Calculate the area of a circle
c - Display a multiplication table
d - Add two numbers
X -exit
please insert your option: x
end of program!
```

Fig107-taske8-B Grade

A Grade

There are two types of variables:

- **Local:** Local variables can be created anywhere in the program's body. For example, if a variable creates in a function's body then it just valid in the function's body and if they be used out of function's body, an error which is 'unknown identifier' will be show.
- **Global:** Global variable must be created before the main function. Then it will be valid in all of program, even in the functions' body. They are very

useful especially when is needed to have more than one result in a function.

Let us to explain more with an example:

```
#include <stdio.h>
void test(void);
int b=5;
int main(void) {
    int a=2;
    printf("%d", a);
    test();
    printf(" %d",b);
    return 0;
}
void test(void)
{
    int a=10;
    printf(" %d",a);
    printf(" %d",b);
}
```

B is a global variable which is valid in all of program. 'a' is a local variable. first 'a' variable is just valid in the main program and the second one is valid just in the function's body. At the beginning 'b' is 5 and 'a' is 2. First printf will be shown 2. Then the program recalls test function. In the body of test function, is created another 'a' variable which is local and is valid just into the function's body. So, with the first printf, 10 will be shown. As 'b' is a global variable, second printf statement in the function's body and the last printf statement in the main program both will be show 5.

```
#include <stdio.h>
void displaymap(void);
void move(void);
int xcoor,ycoor;
char chr;
int main(void)
{
    printf("Please enter an X coordinate (a number between 1 and 10): ");
    scanf("%d",&xcoor);
    printf("Please enter an y coordinate (a number between 1 and 10): ");
    scanf("%d",&ycoor);
    displaymap();
    do
    {
        printf("please enter a character of either 'w','a','s', 'd' or 'x' \n 'w': X should have moved up 1 space\n 'a':X should have moved left 1 space\n 's': then the X should have moved down 1 space \n 'd': X should have moved right 1 space \n x: for exit ");
        while ((chr = getchar()) != '\n' && chr != EOF);
        scanf("%c", &chr);
        if (chr>=65 && chr<=90) chr=chr+32;
        move();
        displaymap();
    }while(chr!='x');
    return 0;
}
```

Fig108-taske8-A Grade

```

void displaymap(void)
{
    int i,j;
    for(i=1;i<=10;i++)
    {
        for(j=1;j<=10;j++)
            if ((xcoor==j)&&(ycoor==i))
                printf(" X ");
            else
                printf(" - ");
            printf("\n");
    }
}

void move(void)
{
    switch (chr)
    {
        case 'w':
        {
            if((ycoor>=2) && (ycoor<=10 ))
                ycoor=ycoor-1;

            else
                ycoor=ycoor;
            break;
        }
        case 'a':
        {
            if ((xcoor>=2)&& (xcoor<=10))
                xcoor=xcoor-1;

```

Fig109-taske8-A Grade

```

        case 's':
        {
            if ((xcoor>=2)&& (xcoor<=10))
                xcoor=xcoor-1;

            else
                xcoor=xcoor;
            break;
        }

        case 's':
        {
            if((ycoor>=1) && (ycoor<=9))
                ycoor=ycoor+1;

            else
                ycoor=ycoor;
            break;
        }

        case 'd':
        {
            if ((xcoor>=1)&& (xcoor<=9))
                xcoor=xcoor+1;

            else
                xcoor=xcoor;
            break;
        }
        case 'x' : {
            break;
        }
        default: printf("Sorry, your input was not recognised, please enter either a, b, c
or d");
    }
}

```

Fig110-taske8-A Grade

```
Please enter an X coordinate (a number between 1 and 10): 5
Please enter an y coordinate (a number between 1 and 10): 5

- - - - -
- - - - -
- - - - -
- - - - -
- - - X -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

please enter a character of either 'w','a','s', 'd' or 'x'
'w': X should have moved up 1 space
'a':X should have moved left 1 space
's': then the X should have moved down 1 space
'd': X should have moved right 1 space
x: for exit w

- - - - -
- - - - -
- - - - -
- - - X -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -
- - - - -

please enter a character of either 'w','a','s', 'd' or 'x'
'w': X should have moved up 1 space
'a':X should have moved left 1 space
's': then the X should have moved down 1 space
'd': X should have moved right 1 space
x: for exit x

- - - - -
- - - - -
- - - - -
- - - X -
- - - - -
- - - - -
- - - - -
```

Fig111-taske8-A Grade

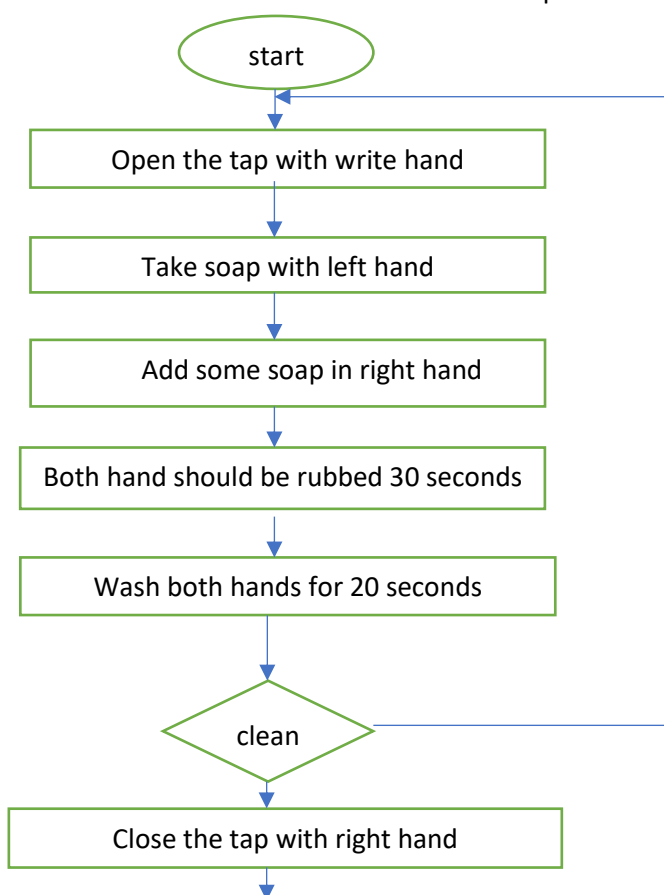
Reflective report

When I started this course, I had not working any programming language for many years. So, this course was very helpful for me as it was a start point to continuing programming. I am really thinking about improving my English language to be a program solving teacher as I understand I really love programming. I am really tanking my teacher who played important role in motivating me. I really do not have any weaknesses in all subjects in this course. My only weakness in programming is that I do not know how to use my science about programming in the real world. If I want to work in a big company as a programmer, they will not ask me to explain an array with 10 items and then with a function find maximum element in the array. I have been in touch with some of my classmates to helping them in programming and I found out that what is their problems is in basic concept of algorithm. As I do not have any weaknesses in this course I just want to say my idea about my classmate's problems.

I believe to getting maximum performance from those students which are new in programming, the best way is starting to teach them concept of algorithm and flowchart before starting to teach them about programming language's statements. It is hard for a new student to write a program which adds two numbers when they do not know anything about algorithm. Also, if a student knows all concepts of algorithm then can start to learn any programming language easily. Teaching concept of algorithm can be started by writing an algorithm for washing hands!! Then the student will know that in any algorithm should be write everything with all the details and step by step.

- 0- Open the tap with write hand.
- 1- Take soap with left hand.
- 2- Add some soap in right hand
- 3- Both hand should be rubbed 30 seconds.
- 4- Wash both hands for 20 seconds.
- 5- Check both hands, if they are clean then go to the first step otherwise go to the next step.
- 6- Close the tap with right hand.

And then the student should learn the concept of flowchart.



Then the next step is teaching very simple algorithms, for example an algorithm for printing sum of two numbers.

1. start
2. get the first number from the user.
3. get the second number from the user.
4. calculate first number+ second number
5. print the result for user

Then the next step is teaching concepts of variables. So, above algorithm should be changed.

1. Start
2. get the first number from the user and insert that into num1.
3. get the first number from the user and insert that into num2.
4. calculate first number+ second number and insert that into num3.
5. Print num3.

The students should be asked to write more than 50 algorithms and then the next step is teaching C language's statements to writing a program. Now their only concern is learning the programming language's statements. The important problem which I have seen between my classmates is that solving their program's errors. It can be either a logic errors or a syntax errors. They must know how to do handy trace in their programs. They can give some entry to their program and then execute the program line by line in a paper. This way is useful for them to finding out their program's logic errors. They must know what is data flow in their programs.

I believe if a student knows different algorithms and solving errors then is possible to learn any programming language.