Contents

Assessment sheet 1

Library stdio.h:

When starting a C program, we need to include libraries that already provide macros, type definitions and functions for tasks that we will need. The first and basic one that we will use in this program will be the standard library, stdio.h (standard inputs and outputs).

Main:

Main is one of the already known functions in c, serves as the starting point for program execution, which means, it is where the computer will read the program. As default, the main function should always return 0 in the end.

Variable:

A variable is a storage area identified by type, which determines the size and layout of the variable's memory, the range of values that can be stored in that memory space and the set of operations that can be applied to the variable. Our program can manipulate the variable value and to make it easier for the user to identify each specific allocated space of memory, we give each variable a different name. A variable are different types of variables:

- Int – (integer) - The most natural size of integer for the machine
- Float- (floating point number) - A single-precision floating point value
- Double – (a bigger Float) - A double-precision floating point value
- Char – (character) A single character from the keyboard
- Void – Represents the absence of type.

Printf:

To print the result of our program or any message to the user we will use the function printf (printf("%'variabletype', 'variablename' list of variables order).

Scanf:

To have the program to read a value from the user we will use the function scanf (scanf("%'variabletype', &'variablename')) which will take the user input and store it in the variable place.

&:

The '&' operator is used to get the address of a variable which means, when we scan a value, we want to store the value in one variable, and for that we get the variable address by with the '&' (example in scanf explanation above). Basically since a variable is a storage area, we are setting that area value as the value scanned.

If statement:

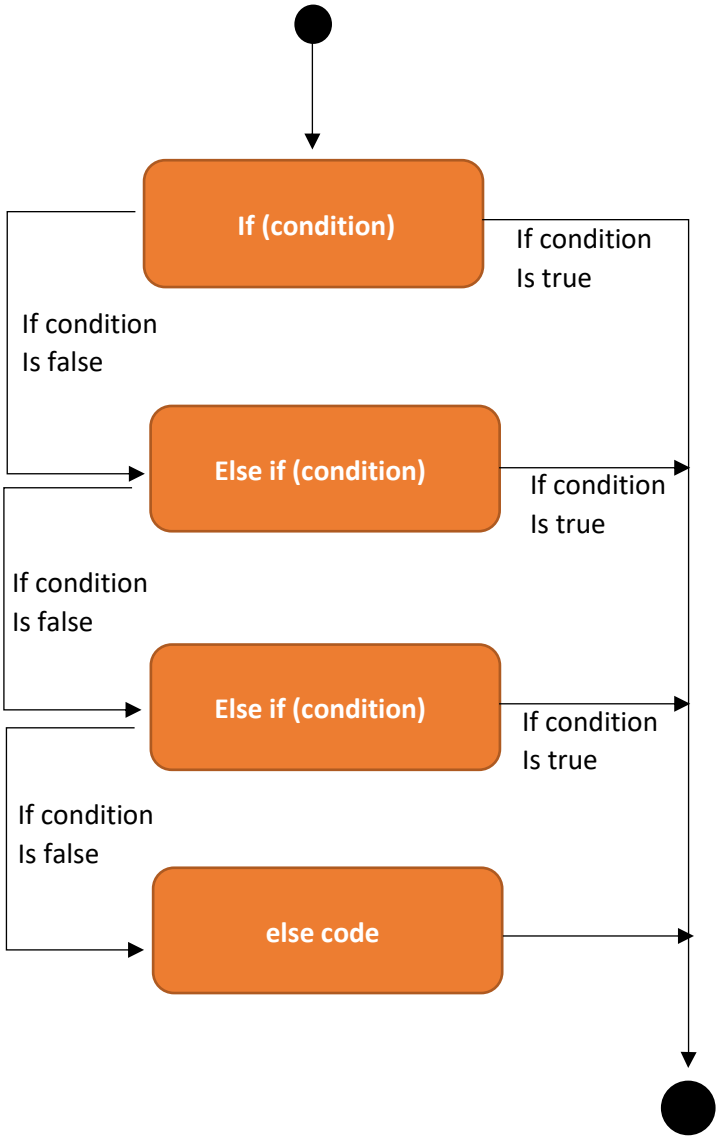
The if statement evaluates the test expression inside the parenthesis, if the expression is evaluated to true, statements inside the body of 'if' are executed and if the test expression is

evaluated to false, statements inside the body of ‘if’ are not executed. For more test expressions we can add ‘else if’ and as last option we can add ‘else’ as something that the program will run if all the test expressions are false.

Clear buffer:

When scanning multiple chars, the program can behave in a different way that we want, after we can the first char and press enter, the most probable thing to happen is that the next char scanned will be fulfilled by “enter” instead of the char we wanted. To prevent this to happen we use the following code after we scan:

```
int ch;

while ((ch = getchar()) != '\n' && ch != EOF);
```



Increment/decrement operators:

Increment operators are used to increase the value of the variable by one and decrement operators are used to decrease the value of the variable by one. For example: If i=1, i++ will be 2, or i- - will be equal to 0.

(List of e.g.)

a+=1 <=> a = a + 1

b-=1 <=> b = b - 1

c *= 1 <=> c = c * 1

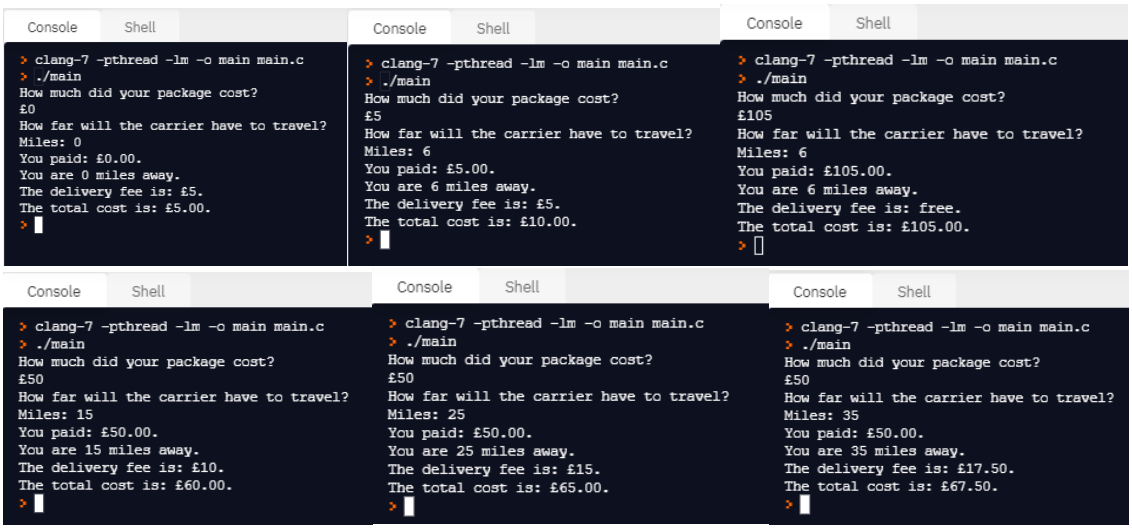
c /= 1 <=> c = c / 1

(D-B)

Code:

```c
#include <stdio.h>

int main(void) {
  //calculate how much a parcel delivery service will charge to deliver goods to a customer
  //This program will take in to account the amount the customer spent and how far (in miles) the carrier will have to travel to deliver the parcel.

  double cost;
  int mileage;

  printf("How much did your package cost?\n£");
  scanf("%lf",&cost);
  printf("How far will the carrier have to travel?\nMiles: ");
  scanf("%d",&mileage);

  /*Delivery is free up to 10 miles, £10 over 10 miles, £15 over 20 miles, £20 over 30 miles
  -cost<=10 miles => free
  -10<mileage<=15 => 10£
  -15<mileage<=20 => 15£
  -20<mileage<=30 => 20£ */
  //We need mileages to be bigger then 0 (this will also give us the 0 as low limit in all coditions and with that we can have only one condition in the other else if)
  //If the mileage is smaller of equal to 10 miles and the value of the goods is over £100 the delivery fee will be free
  if((mileage>=0 && mileage<=10) && cost>100){
    printf("You paid: £%.2lf.\n", cost);
    printf("You are %d miles away.\n", mileage);
    printf("The delivery fee is: free.\n");
    printf("The total cost is: £%.2lf.\n", cost);
  }
  else if((mileage>=0 && mileage<=10) && cost<=100){
    printf("You paid: £%.2lf.\n", cost);
    printf("You are %d miles away.\n", mileage);
    printf("The delivery fee is: £5.\n");
    //We need to add the delivery fee to the cost of the package giving us the total cost for the client
    cost+=5;
    printf("The total cost is: £%.2lf.\n", cost);
  //If the mileage is bigger than 10 and smaller or equal to 15 miles the delivery fee will be £10
  else if(mileage<=20){
    printf("You paid: £%.2lf.\n", cost);
    printf("You are %d miles away.\n", mileage);
    printf("The delivery fee is: £10.\n");
    cost+=10;
    printf("The total cost is: £%.2lf.\n", cost);
  }
  //If the mileage is bigger than 15 and smaller or equal to 20 miles the delivery fee will be £15
  else if(mileage<=30){
    printf("You paid: £%.2lf.\n", cost);
    printf("You are %d miles away.\n", mileage);
    printf("The delivery fee is: £15.\n");
    cost+=15;
    printf("The total cost is: £%.2lf.\n", cost);
  }
  //If the mileage is bigger than 20 miles the delivery fee will be £20
  else{
  double fee = 15 + (mileage-30) * 0.5;
    printf("You paid: £%.2lf.\n", cost);
    printf("You are %d miles away.\n", mileage);
    printf("The delivery fee is: £%.2lf.\n", fee);
    cost+=fee;
    printf("The total cost is: £%.2lf.\n", cost);
  }
  return 0;
}
```

Console:



(A)

Switch statement:

The switch statement allows us to execute one block among many alternatives. The expression provided in the switch statement will be compared with each block case and if they match, the computer will read that block and end when it reads a break instruction. It works the same way as an if statement but should be easier to read and write.

Code:

```c
//Program a simple calculator with '+'. '-', '*' and '/'
#include <stdio.h>

int main(void) {
  char operator;
  double x,y;

  printf("Enter an operator (+, -, *, /): ");
    scanf("%c", &operator);
    printf("Enter two operands: ");
    scanf("%lf %lf",&x, &y);
//We start providing the expression that will be compared if it matches with any case, in this case will be the operator
switch( operator )
{
    //If the operator entered by the user matches with this case (+), it will do x + y
    case '+':
      printf("%.2lf + %.2lf = %.2lf\n",x, y, x+y);
      break;

    //If the operator entered by the user matches with this case (-), it will do x - y
    case '-':
      printf("%.2lf - %.1lf = %.2lf\n",x, y, x-y);
      break;

    //If the operator entered by the user matches with this case (*), it will do x * y
    case '*':
      printf("%.2lf * %.1lf = %.2lf\n",x, y, x*y);
      break;

    //If the operator entered by the user matches with this case (+), it will do x / y
    case '/':
      printf("%.2lf / %.2lf = %.2lf\n",x, y, x/y);
      break;

    //If the user entered a different character then the 4 operators, the program will print a error message
    default:
      printf("Error! Operator is not correct\n");
}
  return 0;
}
```

Console:

```
Console  Shell
> clang-7 -pthread -lm -o main main.c
> ./main
Enter an operator (+, -, *, /): +
Enter two operands: 2 3
2.00 + 3.00 = 5.00
>
```

```
Console  Shell
> clang-7 -pthread -lm -o main main.c
> ./main
Enter an operator (+, -, *, /): -
Enter two operands: 2 3
2.00 - 3.0 = -1.00
>
```

```
Console  Shell
> clang-7 -pthread -lm -o main main.c
> ./main
Enter an operator (+, -, *, /): *
Enter two operands: 2 3
2.00 * 3.0 = 6.00
>
```

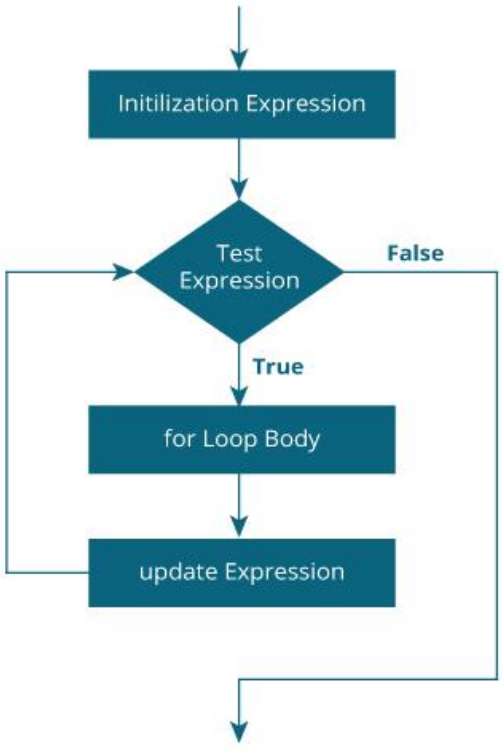
```
Console  Shell
> clang-7 -pthread -lm -o main main.c
> ./main
Enter an operator (+, -, *, /): /
Enter two operands: 2 3
2.00 / 3.00 = 0.67
>
```

Assessment sheet 2

Loops:

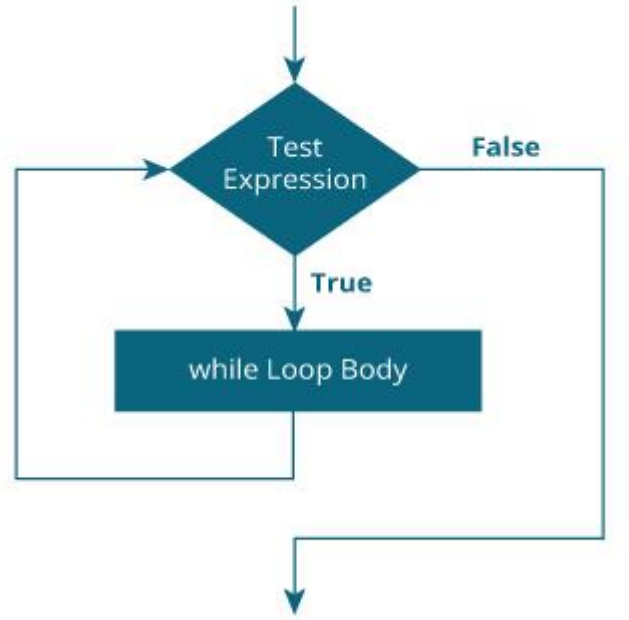
In C we have three types of loops:

1. for loop
    a. The initialization statement is executed only once.
    b. Then, the program evaluates the test expression, if it is evaluated as false the for loop is terminated.
    c. If the test expression is evaluated as true, the statements inside the body of the loop are executed and the update expression is updated.
    d. Step b and c will keep repeating until the test expression is false, terminating the loop.



2. while loop

    a. The while loop evaluates the test expression inside the parenthesis.
    b. If the test expression is true, the statements inside the body of while loop are executed. Then, the test expressions are evaluated again.
    c. The process will continue until it is evaluated as false, ending the loop.

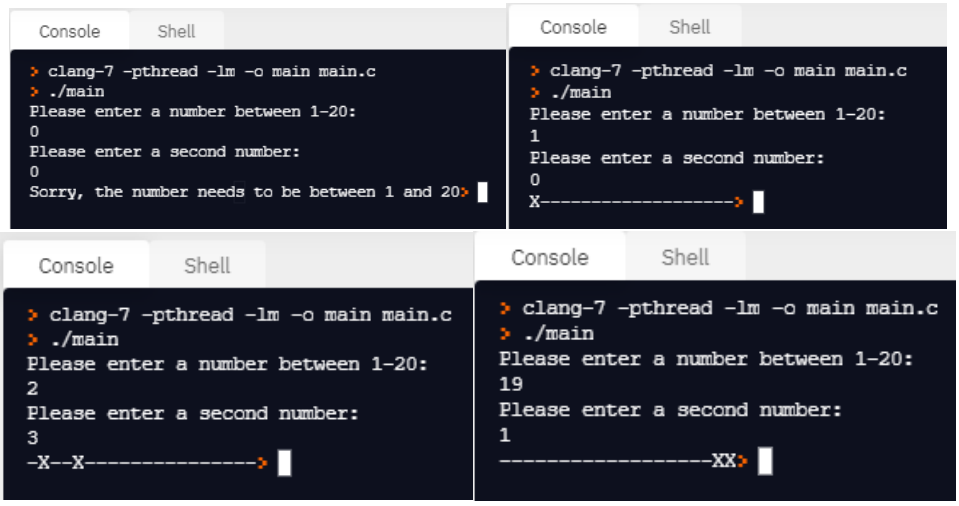


(D-B)

Code:

```c
#include <stdio.h>

int main(void) {
//position1 will set the position for the first 'X'
int position1;
//position2 will set the position for the second 'X',which will be determined by position1+position2
int position2;
//count will determinate the order number of the dash
int count=1;
//Asking the user to choose the position for the first 'X' (Value of position1)
printf("Please enter a number between 1-20:\n");
scanf("%d",&position1);
//Asking the user to choose the position for the second 'X' (Value of position2)
printf("Please enter a second number:\n");
scanf("%d",&position2);
//This if statement will make sure the user types a number between 1 and 20, if not, will show a error message
if(position1 <1 || position1>20){
  printf("Sorry, the number needs to be between 1 and 20");
}
//This else statement will create the dashes line
else{
  while(count<=20){
  //This if statement will switch the dash for an X in position1 and position1+position2 if the position1 and position1
  +position2 match the count number
  if(count==position1 || count==position1+position2){
    printf("X");
    count++;
    }
  //If the 2 'X' positions don't match the count, the program will just print a dash
  else{
    printf("-");
    count++;
  }
}
}
  return 0;
}
```

Console:

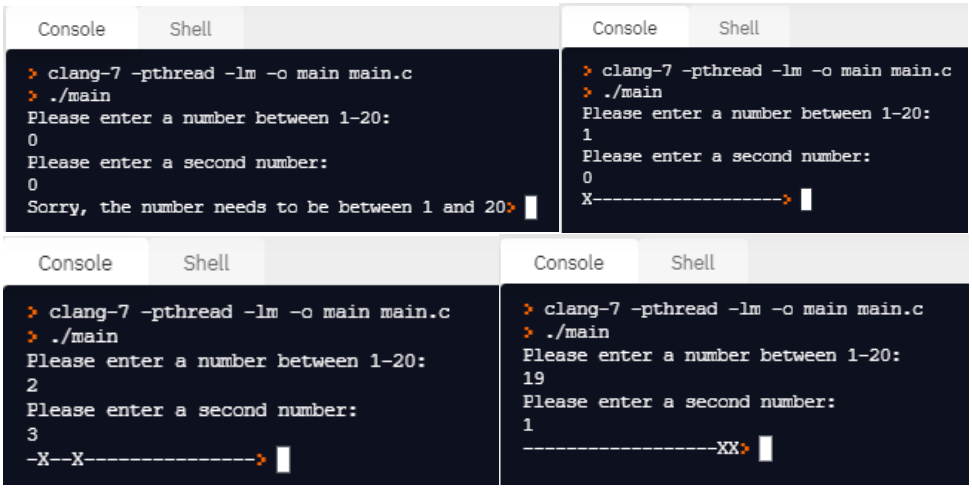


(A)

Code:

```c
#include <stdio.h>

int main(void) {
//position1 will set the position for the first 'X'
int position1;
//position2 will set the position for the second 'X',which will be determined by position1+position2
int position2;
//Variable that will make the incrementation of position
int i;
//Asking the user to choose the position for the first 'X' (Value of position1)
printf("Please enter a number between 1-20:\n");
scanf("%d",&position1);
//Asking the user to choose the position for the second 'X' (Value of position2)
printf("Please enter a second number:\n");
scanf("%d",&position2);
//This if statement will make sure the user types a number between 1 and 20, if not, will show a error message
if(position1 <1 || position1>20){
  printf("Sorry, the number needs to be between 1 and 20");
}
//This else statement will create the dashes line
else{
  //The for cycle will start with position 1, end at position 20 and will go one position forward after each
  interaction
  for(i=1; i<21; i++){
 //This if statement will switch the dash for an X in position1 and position1+position2 if the position1 and
 position1+position2 matches the position number
  if(i==position1 || i==position1+position2){
    printf("X");
    }
  //If the 2 'X' positions don't match the count, the program will just print a dash
  else{
    printf("-");
  }
}
}
  return 0;
}
```

Console:



Assessment sheet 3

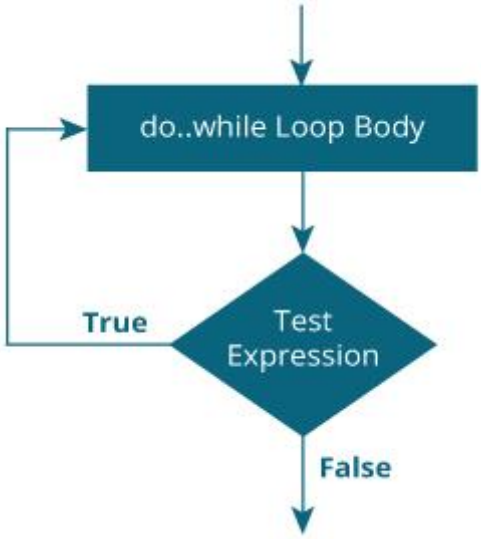
(D-B)

Library math.h:

This library gives us instant access to some more mathematical functions and values, the one we will be using is the value of PI (M_PI). All the functions available in this library take double as an argument and return double as the result.

do…while loop

a. The body of do…while loop is executed once. Only then, the test expression is evaluated.
b. If the test expression is true, the body of the loop is executed again, and the test expression is evaluated.
c. The process will continue until it is evaluated as false, ending the loop.



Code:

```c
#include <stdio.h>
#include <math.h>

int main(void) {
  char letter;
  int ch;
  //While the user doesn't press 'x' or 'X' in the main menu, the program will continue to run
  while(letter!='x' && letter!='X'){
  //Shows the options to user and asks him to choose one
  printf("Please enter the letter which corresponds with your choice:\n");
  printf("a - Calculate the area of a rectangle\n");
  printf("b - Calculate the area of a circle\n");
  printf("c - Display a multiplication table\n");
  printf("d - Add two numbers\n");
  printf("x - Exit program\n");
  scanf("%c", &letter);
  //If the inputed character is different from the ones the program asked for, it will show a error message and ask to try again
  while(letter!='a' && letter!='b' && letter!='c' && letter!= 'd' && letter!='A' && letter!='B' && letter!='C' && letter!= 'D' && letter!= 'x' && letter!= 'X'){

    printf("Sorry, your input was not recognised, please enter either a, b, c or d:\n");
    //This while loop as the objective to clear input buffer
    while ((ch = getchar()) != '\n' && ch != EOF);
    scanf("%c", &letter);
  }
  //Switch statement to select the function that the user choosed
  switch(letter){
    float width, height,radius,firstNumber,secondNumber;
    int tableNumber;
    int i;
    //If the user choosed 'a' or 'A', the program will calculate the area of a rectangle
    case 'a':
    case 'A':
      printf("You have selected - 'a - Calculate the area of a rectangle'\n");
      //Asking for the width of the rectangle
      printf("Please enter the width:\n");
      scanf("%f", &width);
      //Asking for the height of the rectangle
      printf("Please enter the height:\n");
      scanf("%f", &height);
      //Printing the area of the rectangle doing width * height
      printf("The area is: %.2lf\n", width*height);
      while ((ch = getchar()) != '\n' && ch != EOF);
      break;

    case 'b':
    case 'B':
      printf("You have selected - 'b - Calculate the area of a circle'\n");
      //Asking the radius of the circle
      printf("Please enter the radius:\n");
      scanf("%f", &radius);
      //Printing the circle area doing radius * pi
      printf("The area is: %.2lf\n", radius*M_PI);
      while ((ch = getchar()) != '\n' && ch != EOF);
      break;

    case 'c':
    case 'C':
      printf("You have selected - 'c - Display a multiplication table'\n");
      //Asking the number of the multiplication table
      printf("Please enter the a number:\n");
      scanf("%d", &tableNumber);
      //Printing the multiplication table for the number given by the user.
      printf("The multiplication table for %d is:\n", tableNumber);
      //The cicle for will multiply the given number for 1 to 10
      for(i=1;i<11; i++){
        printf("%d x %d = %d\n",i, tableNumber, tableNumber*i);
      }
      while ((ch = getchar()) != '\n' && ch != EOF);
      break;

    case 'd':
    case 'D':
      printf("You have selected - 'd - Add two numbers'\n");
      //Asking the first number
      printf("Please enter the first number:\n");
      scanf("%f", &firstNumber);
      //Asking the second number
      printf("Please enter the second number:\n");
      scanf("%f", &secondNumber);
      //Printing the addition result form the first and second number
      printf("The addition from %.2lf and %.2lf is %.2lf.\n", firstNumber, secondNumber, firstNumber+secondNumber);
      while ((ch = getchar()) != '\n' && ch != EOF);
      break;
  }
  }
  return 0;
}
```

Console:

Console Shell

```
> 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 program
g
Sorry, your input was not recognised, please enter either a, b, c or d:
A
You have selected - 'a - Calculate the area of a rectangle'
Please enter the width:
2
Please enter the height:
3
The area is: 6.00
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 program
2
Sorry, your input was not recognised, please enter either a, b, c or d:
b
You have selected - 'b - Calculate the area of a circle'
Please enter the radius:
3
The area is: 9.42
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 program
```

Console Shell

```
> 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 program
c
You have selected - 'c - Display a multiplication table'
Please enter the a number:
4
The multiplication table for 4 is:
1 x 4 = 4
2 x 4 = 8
3 x 4 = 12
4 x 4 = 16
5 x 4 = 20
6 x 4 = 24
7 x 4 = 28
8 x 4 = 32
9 x 4 = 36
10 x 4 = 40
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 program
d
You have selected - 'd - Add two numbers'
Please enter the first number:
4
Please enter the second number:
5
The addition from 4.00 and 5.00 is 9.00.
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 program
x
>
```

(A)

Code:

```c
#include <stdio.h>
#include <math.h>

int main(void) {
  char letter;
  int ch;
//This do starts the "do...while" loop, so the program will start running "do" and will repeat it untill the while condition is true ( in this case, until the user presses X whilst on menu)
do{
  //Shows the options to user and asks him to choose one
  printf("Please enter the letter which corresponds with your choice:\n");
  printf("a - Calculate the area of a rectangle\n");
  printf("b - Calculate the area of a circle\n");
  printf("c - Display a multiplication table\n");
  printf("d - Add two numbers\n");
  printf("x - Exit program\n");
  scanf("%c", &letter);

//If the inputed character is different from the ones the program asked for, it will show a error message and ask to try again
  while(letter!='a' && letter!='b' && letter!='c' && letter!= 'd' && letter!='A' && letter!='B' && letter!='C' && letter!= 'D' && letter!= 'x' && letter!= 'X'){

    printf("Sorry, your input was not recognised, please enter either a, b, c or d:\n"); while ((ch = getchar()) != '\n' && ch != EOF);
    scanf("%c", &letter);
  }

//Switch statement to select the function that the user choosed
  switch(letter){
    float width, height,radius,firstNumber,secondNumber;
    int tableNumber;
    int i;

    //If the user choosed 'a' or 'A', the program will calculate the area of a rectangle
    case 'a':
    case 'A':
      printf("You have selected - 'a - Calculate the area of a rectangle'\n");
      //Asking for the width of the rectangle
      printf("Please enter the width:\n");
      scanf("%f", &width);
      //Asking for the height of the rectangle
      printf("Please enter the height:\n");
      scanf("%f", &height);
      //Printing the area of the rectangle doing width * height
      printf("The area is: %.2lf\n", width*height);
      //This while loop as the objective to clear input buffer
      while ((ch = getchar()) != '\n' && ch != EOF);
      break;

    case 'b':
    case 'B':
      printf("You have selected - 'b - Calculate the area of a circle'\n");
      //Asking the radius of the circle
      printf("Please enter the radius:\n");
      scanf("%f", &radius);
      //Printing the circle area doing radius * pi
      printf("The area is: %.2lf\n", radius*M_PI);
      while ((ch = getchar()) != '\n' && ch != EOF);
      break;

    case 'c':
    case 'C':
      printf("You have selected - 'c - Display a multiplication table'\n");
      //Asking the number of the multiplication table
      printf("Please enter the a number:\n");
      scanf("%d", &tableNumber);
      //Printing the multiplication table for the number given by the user.
      printf("The multiplication table for %d is:\n", tableNumber);
      //The cicle for will multiply the given number for 1 to 10
      for(i=1;i<11; i++){
        printf("%d x %d = %d\n",i, tableNumber, tableNumber*i);
      }
      while ((ch = getchar()) != '\n' && ch != EOF);
      break;
```

```
72
73      case 'd':
74      case 'D':
75        printf("You have selected - 'd - Add two numbers'\n");
76        //Asking the first number
77        printf("Please enter the first number:\n");
78        scanf("%f", &firstNumber);
79        //Asking the second number
80        printf("Please enter the second number:\n");
81        scanf("%f", &secondNumber);
82        //Printing the addition result form the first and second number
83        printf("The addition from %.2lf and %.2lf is %.2lf.\n", firstNumber, secondNumber, firstNumber+secondNumber);
84        while ((ch = getchar()) != '\n' && ch != EOF);
85        break;
86    }
87    }
88    //If this condition is true, the program will end this cicle, if its false, it will run again starting in "do"
89      while(letter!='x' && letter!='X');
90    return 0;
91  }
```

Console:

Console Shell

```
> 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 program
j
Sorry, your input was not recognised, please enter either a, b, c or d:
A
You have selected - 'a - Calculate the area of a rectangle'
Please enter the width:
4
Please enter the height:
5
The area is: 20.00
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 program
B
You have selected - 'b - Calculate the area of a circle'
Please enter the radius:
5
The area is: 15.71
```

```
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 program
c
You have selected - 'c - Display a multiplication table'
Please enter the a number:
9
The multiplication table for 9 is:
1 x 9 = 9
2 x 9 = 18
3 x 9 = 27
4 x 9 = 36
5 x 9 = 45
6 x 9 = 54
7 x 9 = 63
8 x 9 = 72
9 x 9 = 81
10 x 9 = 90
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 program
d
You have selected - 'd - Add two numbers'
Please enter the first number:
7
Please enter the second number:
8
The addition from 7.00 and 8.00 is 15.00.
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 program
x
>
```

Assessment sheet 4

Library stdlib.h:

We used stdlib.h to get the absolute value from an operation. This header includes four variables and different macros and functions.

The variables:

size_t : This is the unsigned integral type and is the result of the sizeof keyword.

wchar_t: This is an integer type of the size of a wide character constant.

div_t: This is the structure returned by the div function.

idiv_t: This is the structure returned by the ldiv function.

Library ctype.h:

We used ctype library to get all scanned chars in capitals with the function “toupper”. This library is useful for testing and mapping characters.

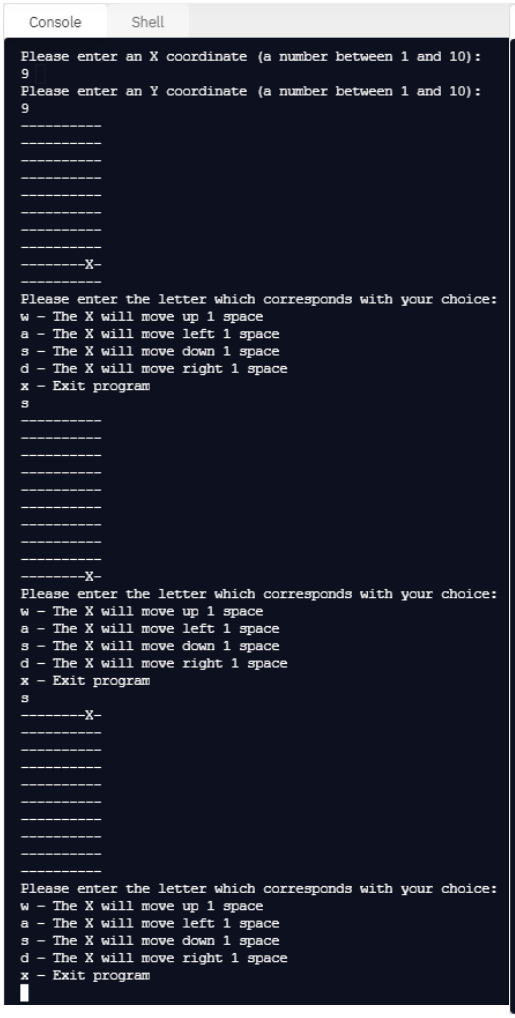
(D-B)

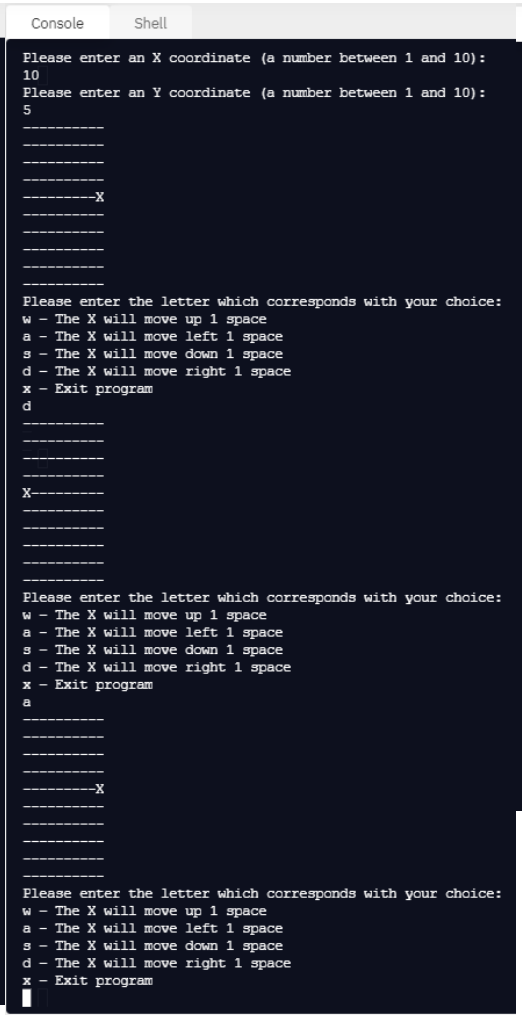
Code:

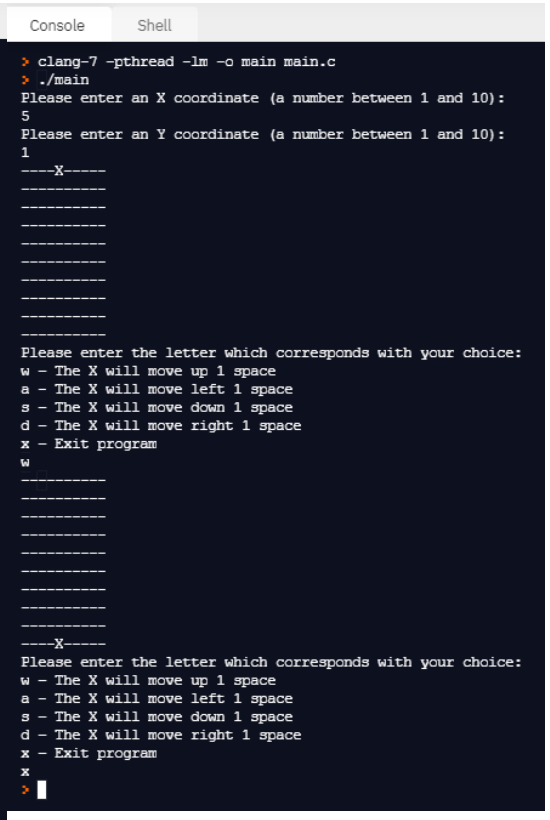
```
46
47      printf("Sorry, your input was not recognised, please enter either w, a, s, d or x:\n");
48      scanf("%c", &letter);
49      while ((ch = getchar()) != '\n' && ch != EOF);
50    }
51    //If the user presses 'a' program will decrement 1 to x, making the 'X' move left 1 space
52    if(letter=='a' || letter=='A'){
53      x-=1;
54    }
55    //If the user presses 'd' program will increment 1 to x, making the 'X' move right 1 space
56    else if(letter=='d' || letter=='D'){
57      x+=1;
58    }
59    //If the user presses 'w' program will decrement 1 to x, making the 'X' move up 1 space
60    else if(letter=='w' || letter=='W'){
61      y-=1;
62    }
63    //If the user presses 'a' program will increment 1 to x, making the 'X' move down 1 space
64    else if(letter=='s' || letter=='S'){
65      y+=1;
66    }
67    //In case the x variable reaches 0 it will change it's value to 10 in order to make the 'X' appear on the
      opposite side of the grid
68    if(x==0){
69      x=10;
70    }
71    //In case the x variable reaches a a value above 10 , it will give an absolute(positive) value between 1 to
      10, the purpose it that when the x reaches 11, we will have 11/10=1, and with that we get the 'X' appear on
      the opposite side of the grid
72    else if(x>10){
73      x=abs(x/10);
74    }
75    //In case the y variable reaches 0 it will change it's value to 10 in order to make the 'X' appear on the
      opposite side of the grid
76    if(y==0){
77      y=10;
78    }
79    //In case the y variable reaches a a value above 10 , it will give an absolute(positive) value between 1 to
      10, the purpose it that when the y reaches 11, we will have 11/10=1, and with that we get the 'X' appear on
      the opposite side of the grid
80    else if(y>10){
81      y=abs(y/10);
82    }
83    }
84    //Thise while loop will make the program run untill the user presses 'x' when on the menu
85    while(letter!='x' && letter!='X');
86    return 0;
87  }
```

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

int main(void) {
  int x,y;
  int i,j;
  int ch;
  char letter;


  printf("Please enter an X coordinate (a number between 1 and 10):\n");
  scanf("%d", &x);
  printf("Please enter an Y coordinate (a number between 1 and 10):\n");
  scanf("%d", &y);
  while ((ch = getchar()) != '\n' && ch != EOF);
  do{
    //This for loop is responsible for rows
    for(j=1; j<11; j++){
      //This for loop is responsible for columns
      for(i=1; i<11; i++){
        //This if statement will put an 'X' if the "coordinates" matches the ones scanned from the user
        if(i==x && j==y){
          printf("X");
        }
        //This else prints the dashes
        else{
        printf("-");
        }
  }
  //This printf will change the line after the 10 dashes in a raw
  printf("\n");
  }
  printf("Please enter the letter which corresponds with your choice:\n");
  printf("w - The X will move up 1 space\n");
  printf("a - The X will move left 1 space\n");
  printf("s - The X will move down 1 space\n");
  printf("d - The X will move right 1 space\n");
  printf("x - Exit program\n");
  //This scanf will get the variable responsible to "move" the 'X'
  scanf("%c",&letter);

    //This while loop as the objective to clear input buffer
   while ((ch = getchar()) != '\n' && ch != EOF);

  while(letter!='w' && letter!='a' && letter!='s' && letter!= 'd'  && letter!='W' && letter!='A' &&
  letter!='S' && letter!= 'D' && letter!= 'x' && letter!= 'X'){
```

Console:


Console
Shell
Please enter an X coordinate (a number between 1 and 10):
9
Please enter an Y coordinate (a number between 1 and 10):
9
--------X-
Please enter the letter which corresponds with your choice:
w - The X will move up 1 space
a - The X will move left 1 space
s - The X will move down 1 space
d - The X will move right 1 space
x - Exit program
s
--------X-
Please enter the letter which corresponds with your choice:
w - The X will move up 1 space
a - The X will move left 1 space
s - The X will move down 1 space
d - The X will move right 1 space
x - Exit program
s
--------X-
Please enter the letter which corresponds with your choice:
w - The X will move up 1 space
a - The X will move left 1 space
s - The X will move down 1 space
d - The X will move right 1 space
x - Exit program


Console
Shell
Please enter an X coordinate (a number between 1 and 10):
10
Please enter an Y coordinate (a number between 1 and 10):
5
---------X
Please enter the letter which corresponds with your choice:
w - The X will move up 1 space
a - The X will move left 1 space
s - The X will move down 1 space
d - The X will move right 1 space
x - Exit program
d
X---------
Please enter the letter which corresponds with your choice:
w - The X will move up 1 space
a - The X will move left 1 space
s - The X will move down 1 space
d - The X will move right 1 space
x - Exit program
a
---------X
Please enter the letter which corresponds with your choice:
w - The X will move up 1 space
a - The X will move left 1 space
s - The X will move down 1 space
d - The X will move right 1 space
x - Exit program


Console
Shell
> 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):
1
----X-----
Please enter the letter which corresponds with your choice:
w - The X will move up 1 space
a - The X will move left 1 space
s - The X will move down 1 space
d - The X will move right 1 space
x - Exit program
w
----X-----
Please enter the letter which corresponds with your choice:
w - The X will move up 1 space
a - The X will move left 1 space
s - The X will move down 1 space
d - The X will move right 1 space
x - Exit program
x
>

(A)

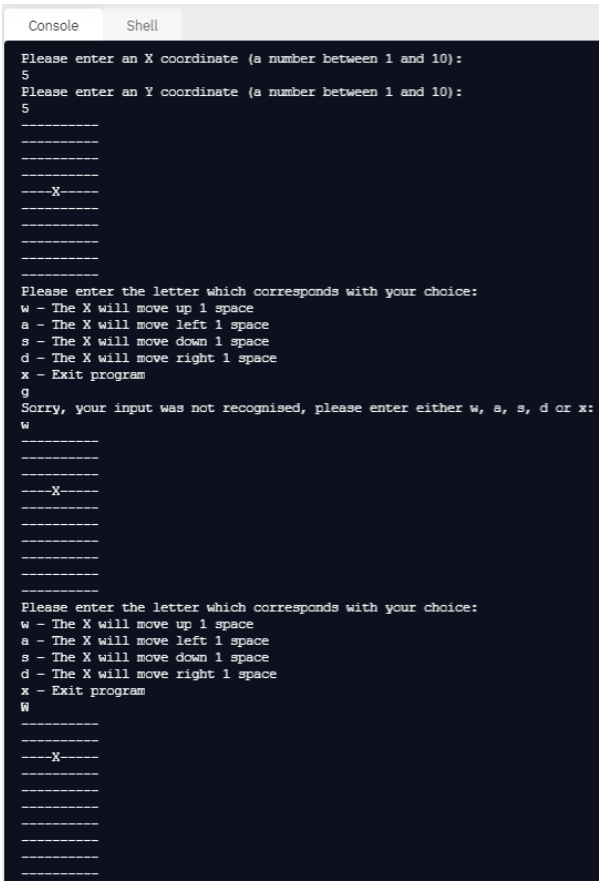
Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main(void) {
  int x,y;
  int i,j;
  int ch;
  char letter;
  //Asking the user to choose the first place to have the "X"
  printf("Please enter an X coordinate (a number between 1 and 10):\n");
  scanf("%d", &x);
  printf("Please enter an Y coordinate (a number between 1 and 10):\n");
  scanf("%d", &y);
  do{
    //This for loop is responsible for rows
    for(j=1; j<11; j++){
      //This for loop is responsible for columns
      for(i=1; i<11; i++){
        //This if statement will put an 'X' if the "coordinates" matches the ones scanned from the user
        if(i==x && j==y){
          printf("X");
        }
        //This else prints the dashes
        else{
        printf("-");
        }
  }
  //This printf will change the line after the 10 dashes in a row
  printf("\n");
  }
  printf("Please enter the letter which corresponds with your choice:\n");
  printf("w - The X will move up 1 space\n");
  printf("a - The X will move left 1 space\n");
  printf("s - The X will move down 1 space\n");
  printf("d - The X will move right 1 space\n");
  printf("x - Exit program\n");
  //This scanf will get the variable responsible to "move" the 'X'
  scanf("%c",&letter);
  //This while loop as the objective to clear input buffer
  while ((ch = getchar()) != '\n' && ch != EOF);
  while(letter!='w' && letter!='a' && letter!='s' && letter!= 'd'  && letter!='W' && letter!='A' && letter!='S' &&
  letter!= 'D' && letter!= 'x' && letter!= 'X'){

    printf("Sorry, your input was not recognised, please enter either w, a, s, d or x:\n");
    scanf("%c", &letter);
    while ((ch = getchar()) != '\n' && ch != EOF);
  }
  //toupper function will transform the letter scanned in uppercase, in this case there is no need to have an or
  condition in if statements
  letter = toupper(letter);
  //If the user presses 'a' program will decrement 1 to x, making the 'X' move left 1 space
  if(letter=='A'){
    x-=1;
  }
  //If the user presses 'd' program will increment 1 to x, making the 'X' move right 1 space
  else if(letter=='D'){
    x+=1;
  }
  //If the user presses 'w' program will decrement 1 to x, making the 'X' move up 1 space
  else if(letter=='W'){
    y-=1;
  }
  //If the user presses 'a' program will increment 1 to x, making the 'X' move down 1 space
  else if(letter=='S'){
    y+=1;
```

```
}
//In case the x variable reaches 0 it will change it's value to 10 in order to make the 'X' appear on the
opposite side of the grid
if(x==0){
  x=10;
}
//In case the x variable reaches a a value above 10 , it will give an absolute(positive) value between 1 to
10, the purpose it that when the x reaches 11, we will have 11/10=1, and with that we get the 'X' appear on
the opposite side of the grid
else if(x>10){
  x=abs(x/10);
}
//In case the y variable reaches 0 it will change it's value to 10 in order to make the 'X' appear on the
opposite side of the grid
if(y==0){
  y=10;
}
//In case the y variable reaches a a value above 10 , it will give an absolute(positive) value between 1 to
10, the purpose it that when the y reaches 11, we will have 11/10=1, and with that we get the 'X' appear on
the opposite side of the grid
else if(y>10){
  y=abs(y/10);
}
}
//Thise while loop will make the program run untill the user presses 'x' when on the menu
while(letter!='x' && letter!='X');
return 0;
}
```

I used “toupper” function so that all scanned chars (letters) would become uppercase, in this way I did not need to use an or condition in If statements.

Console:



Assessment sheet 5

Array:

To store multiple values in one variable we use arrays, which the compiler will allocate memory after initializing the array and save that memory space for that array elements.

To initialize an array we can either declare the array size doing for example 'int array[5]' , or we can declare its elements and the compiler will automatically know the space needed for the array. (int array [] = {1, 3, 5, 6}, the array will be size 4).

Terminator:

In cases that we have big arrays, and we do not know the size of it, we can add what it is called a terminator, in the first position after the last position used of the array, we had a value that would not be used in the array, in an all-positive value array we could add a -1. In this case when doing loops, we can look to end the loop when the loop matches the value -1.

e.g.:

```
int array[ ] = { 3, 5, 1, 6, -1};

while(array[count] != -1){

(…code…)

}
```

sizeOf:

This function is used to get the size of the operant. It can be applied to all data (integers, floats, doubles,etc)

(D-B)

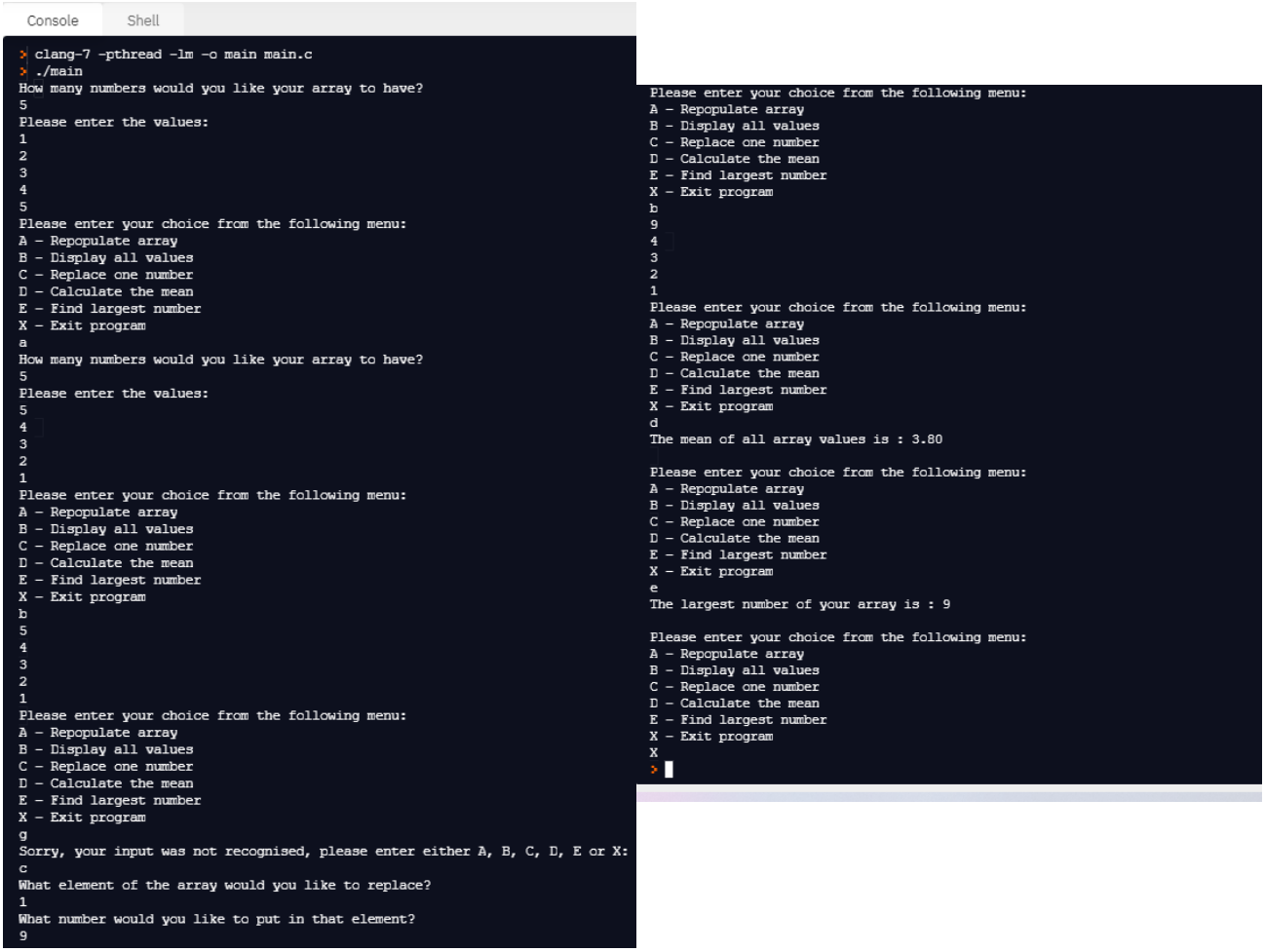
Code:

main.c

```c
#include <stdio.h>
#include <ctype.h>

int main(void) {
  char letter;
  int ch;
  int sizeArray;
  int count;
  int array[100];

  //First we create our starting array
  printf("How many numbers would you like your array to have?\n");
  scanf("%d", &sizeArray);
  printf("Please enter the values:\n");
  while(count<sizeArray){
    scanf("%d", &array[count]);
    count++;
  }
  //We set the terminator after the last scanned array position
  array[sizeArray]=-1;
  //do...while loop, to keep running the menu so that the user can keep using the program
  do{
    printf("Please enter your choice from the following menu:\n");
    printf("A - Repopulate array\n");
    printf("B - Display all values\n");
    printf("C - Replace one number\n");
    printf("D - Calculate the mean\n");
    printf("E - Find largest number\n");
    printf("X - Exit program\n");
    //Clear the buffer
    while ((ch = getchar()) != '\n' && ch != EOF);
    scanf("%c", &letter);
    //Put the char scanned in uppercase to reduce conditions in if statements
    letter = toupper(letter);
    //If the inputed character is different from the ones the program asked for, it will show a error
    message and ask to try again
    while(letter!='A' && letter!='B' && letter!='C' && letter!= 'D' && letter!= 'E' && letter!= 'x' &&
    letter!= 'X'){

    printf("Sorry, your input was not recognised, please enter either A, B, C, D, E or X:\n");
     while ((ch = getchar()) != '\n' && ch != EOF);
    scanf("%c", &letter);
    letter = toupper(letter);
  }
  //Creates a new array
  if(letter=='A'){
    count=0;
    printf("How many numbers would you like your array to have?\n");
    scanf("%d", &sizeArray);
```

```
    count=0;
    printf("How many numbers would you like your array to have?\n");
    scanf("%d", &sizeArray);
    printf("Please enter the values:\n");
    while(count<sizeArray){
    scanf("%d", &array[count]);
    count++;
  }
  array[sizeArray]=-1;
  }
  //Prints the array
  else if(letter=='B'){
    for(int i=0; i<sizeArray; i++){
      printf("%d\n",array[i]);
    }
  }
  //Replaces a specific element of the array
  else if(letter=='C'){
    int replace;
    int value;
    printf("What element of the array would you like to replace?\n");
    scanf("%d", &replace);
    while(replace>sizeArray){
    printf("Sorry, your input was not recognised, please enter a number smaller than %d:\n", sizeArray+1);
    scanf("%d", &replace);
    }
    printf("What number would you like to put in that element?\n");
    scanf("%d", &value);
    array[replace-1] = value;
  }
  //Calculates the mean of the array values
  else if(letter=='D'){
    float result=0;
    for(int i=0; i<sizeArray; i++){
      result+= array[i];
    }
  printf("The mean of all array values is : %.2lf\n\n",result=result/sizeArray);
  }
  //Finds the largest number in the array
  else if(letter=='E'){
    int result;
    //This for cycle will compare the element 0 with the i, will store the largest number in the first position of
    the array, and will now compare the element 0 with the i+1, in the end we will have the largest number in the
    position 0
    for(int i=0; i<sizeArray; i++){
      if(array[0]<array[i]){
        array[0] = array[i];
      }
    }
  printf("The largest number of your array is : %d\n\n", array[0]);
    }
  }
  //This ends the loop if the user enters X whilst on the menu
  while(letter!='x' && letter!='X');
    return 0;
}
```

Console:



(A)

Different approach: Instead of using a terminator, we can check for the number of elements of the array by doing:

n = sizeof(array);

printf("The size of the array is %d", n/sizeof(int));

In this program we asked the user the number of elements that they wanted to have in the array, and with that we got the number of elements of the array. But if didn't ask the user for that, and instead, we just asked him for the elements that he wanted to have inside the array, with that expression we could get the size of the array and we could use that as a limit in the loops.

Getchar:

This function returns the character entered in the console by the user. Instead of doing a scanf and store the char value we can just use getchar and store the value to unassigned "name".

We can see in the code that instead of having to clear the buffer and have a scanf we can just have a getchar instead.

```c
do{
  printf("Please enter your choice from the following menu:\n");
  printf("A - Repopulate array\n");
  printf("B - Display all values\n");
  printf("C - Replace one number\n");
  printf("D - Calculate the mean\n");
  printf("E - Find largest number\n");
  printf("X - Exit program\n");

  letter = getchar();
```

```c
do{
  printf("Please enter your choice from the following menu:\n");
  printf("A - Repopulate array\n");
  printf("B - Display all values\n");
  printf("C - Replace one number\n");
  printf("D - Calculate the mean\n");
  printf("E - Find largest number\n");
  printf("X - Exit program\n");
  //Clear the buffer
  while ((ch = getchar()) != '\n' && ch != EOF);
  scanf("%c", &letter);
```

Assessment sheet 6

In this program we used a temporary variable, we called it “temp” and we used it to switch array elements.

(D-B)

```c
#include <stdio.h>

int main(void) {
  int array[10];
  int count=0;
  int i=0;

  array[0]=6;
  array[1]=5;
  array[2]=3;
  array[3]=1;
  array[4]=2;
  array[5]=-1;
//This loop will print our starting array
while(array[count] != -1){
  printf("%d\n", array[count]);
    count++;
  }
  printf("\n");
  count = 0;
//This first while loop will increment the first comparable array value, the element "i"
while(array[i] != -1){
  count = i+1;
  //This second while loop will find the smallest value and put in in the element "i"
  while(array[count] != -1){
    int temp;
    if(array[i]>array[count]){
      temp=array[i];
      array[i]=array[count];
      array[count]=temp;
    }
    count++;
  }
  i++;
  }

  count=0;
//This loop will print our array after sorting
  while(array[count] != -1){
  printf("%d\n", array[count]);
    count++;
  }
  return 0;
}
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
6
5
3
1
2

1
2
3
5
6
>
```

(A)

```c
#include <stdio.h>

int main(void) {
  int array[10];
  int count=0;
  int i=0;

  array[0]=6;
  array[1]=5;
  array[2]=3;
  array[3]=1;
  array[4]=2;
  array[5]=-1;
//This loop will print our starting array
while(array[count] != -1){
  printf("%d\n", array[count]);
    count++;
  }
  printf("\n");
  count = 0;

for(int i; i < 5; i++){
  int key = array[i];
  int j = i-1;

  while(key < array[j] && j>=0) {
    array[j +1] = array[j];
    --j;
  }
  array[j+1] = key;
  }

count=0;
//This loop will print our array after sorting
  while(array[count] !=-1){
  printf("%d\n", array[count]);
    count++;
  }
  return 0;
}
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
6
5
3
1
2

1
2
3
5
6
>
```

As a different way to sort an array of integers I choose the insertion sort algorithm, which follow the next steps:

1. We assume the first element as sorted, taking the second element of the array and store it separately in “key”. Then we compare the “key” with the first element and if the first element is greater than “key”, the “key” is placed before the first element.
2. We compare the third element with the elements on the left placing it behind the element smaller than itself. If there is no smaller element, then we just place it in the beginning of the array.
3. Repeat step 2 until all array is sorted.

Assessment sheet 7

Functions:

A function is a block of code that runs using parameters (variables scanned or obtained while the program in running) to do computing and usually returns some value in the end even though it could return nothing and show some outputs instead. Usually, it is used to organize code and to avoid having redundant lines of coding, since we can just keep calling the same function when needed instead of having to write all over again. Besides main which is the only function that automatically runs in the compiler, all other functions outside main will have to be called in main, so the program will not have to run all functions unless we need them.

Global variables:

Usually, we declare variables inside the function and use it as we need and after running the function the variable will not exist in the rest of the program, but if we want to use the same variable all along the program, we can declare it outside the functions and use it when we need it. We can set a value when starting the variable, but the predominant value will be the one set inside the function.

(D-B)

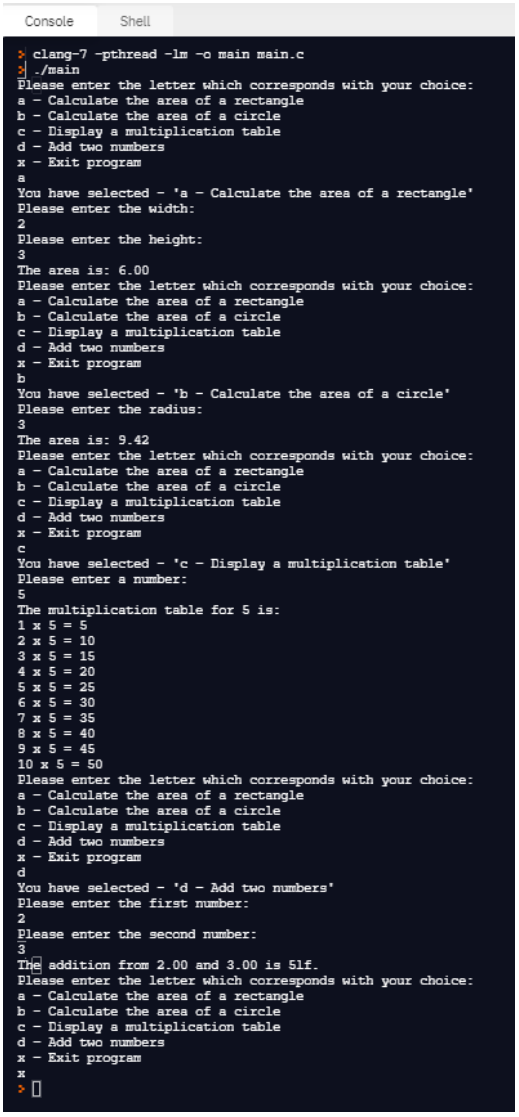
Code:

main.c

```c
#include <stdio.h>
#include <math.h>

//This funcion calculates the area of a rectangle
float areaRectangle(float width, float height){
  float result;
  result = width*height;
  return result;
}

//This funcion calculates the area of a circle
float areaCircle(float radius){
  float result;
  result = radius*M_PI;
  return result;
}

//This funcion multiplies 2 numbers
int multiplyTwoNumbers(int numberOne, int numberTwo){
  int result = numberOne*numberTwo;
  return result;
}

//This funcion prints out a multiplication table from 1 to 10
void multiplicationTable(int tableNumber){
int i;
printf("The multiplication table for %d is:\n", tableNumber);
  for(i=1;i<11; i++){
    int result = multiplyTwoNumbers(tableNumber, i);
    printf("%d x %d = %d\n",i, tableNumber, result);
  }
}


//This funciion adds 2 numbers
int addNumbers(float firstNumber, float secondNumber){
  float result = firstNumber + secondNumber;
  return result;
}

int main(void) {
  char letter;
  int ch;
  float width, height;

//This do starts the "do...while" loop, so the program will start running "do" and will repeat it untill the while condition is true ( in this case, until the user presses X whilst on menu)
do{
  //Shows the options to user and asks him to choose one
  printf("Please enter the letter which corresponds with your choice:\n");
  printf("a - Calculate the area of a rectangle\n");
  printf("b - Calculate the area of a circle\n");
  printf("c - Display a multiplication table\n");
  printf("d - Add two numbers\n");
  printf("x - Exit program\n");
  scanf("%c", &letter);

  //If the inputed character is different from the ones the program asked for, it will show a error message and ask to try again
  while(letter!='a' && letter!='b' && letter!='c' && letter!= 'd' && letter!='A' && letter!='B' && letter!='C' && letter!= 'D' && letter!= 'x' && letter!= 'X'){
```

```
59
60      printf("Sorry, your input was not recognised, please enter either a, b, c or d:\n"); while ((ch = getchar()) != '\n' && ch != EOF);
61      scanf("%c", &letter);
62    }
63    //Switch statement to select the function that the user choosed
64    switch(letter){
65      float width, height,radius,firstNumber,secondNumber;
66      int tableNumber;
67      int i;
68
69      //If the user choosed 'a' or 'A', the program will calculate the area of a rectangle
70      case 'a':
71      case 'A':
72
73        printf("You have selected - 'a - Calculate the area of a rectangle'\n");
74        //Asking for the width of the rectangle
75        printf("Please enter the width:\n");
76        scanf("%f", &width);
77        //Asking for the height of the rectangle
78        printf("Please enter the height:\n");
79        scanf("%f", &height);
80        //Get the result from the function areaRectangle using the parameters scaned
81        float areaRec = areaRectangle(width, height);
82        //Print the result from function areaRectangle
83        printf("The area is: %.2lf\n", areaRec);
84        //This while loop as the objective to clear input buffer
85        while ((ch = getchar()) != '\n' && ch != EOF);
86        break;
87
88      case 'b':
89      case 'B':
90
91        printf("You have selected - 'b - Calculate the area of a circle'\n");
92        //Asking the radius of the circle
93        printf("Please enter the radius:\n");
94        scanf("%f", &radius);
95        //Get the result from the function areaCircle using the parameters scaned
96        float areaCir = areaCircle(radius);
97        //Print the result from function areaCircle
98        printf("The area is: %.2lf\n", areaCir);
99
100       while ((ch = getchar()) != '\n' && ch != EOF);
101       break;
102
103     case 'c':
104     case 'C':
105
106       printf("You have selected - 'c - Display a multiplication table'\n");
107       //Asking the number of the multiplication table
108       printf("Please enter a number:\n");
109       scanf("%d", &tableNumber);
110       //This will call the funcion and the compiller will run the function, in this case it will not return a value, instead the function will do
          consecutive printfs providing the multiplication table of the number given
111       multiplicationTable(tableNumber);
112
113       while ((ch = getchar()) != '\n' && ch != EOF);
114       break;
115
116     case 'd':
117     case 'D':
118
119       printf("You have selected - 'd - Add two numbers'\n");
120       //Asking the first number
121       printf("Please enter the first number:\n");
122       scanf("%f", &firstNumber);
123       //Asking the second number
124       printf("Please enter the second number:\n");
125       scanf("%f", &secondNumber);
126       //Get the result from the function addNumbers using the parameters scaned
127       int add = addNumbers(firstNumber, secondNumber);
128       //Print the result from function addNumbers
129       printf("The addition from %.2lf and %.2lf is %dlf.\n", firstNumber, secondNumber, add);
130
131       while ((ch = getchar()) != '\n' && ch != EOF);
132       break;
133   }
134   }
135    //If this condition is true, the program will end this cicle, if its false, it will run again starting in "do"
136     while(letter!='x' && letter!='X');
137   return 0;
138 }
```

Console:



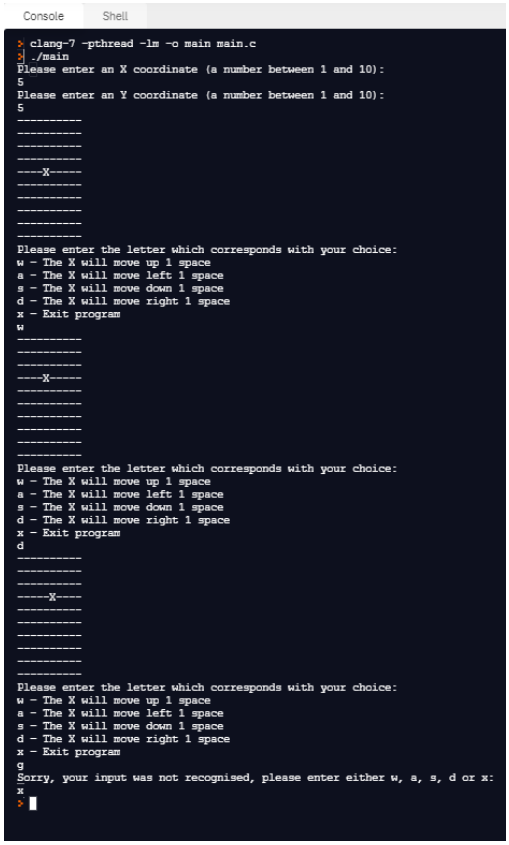
(A)

Code:

main.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>

int i,j,x,y,ch;
char letter;

//This function will display the map on the console and place the 'X' where the user wants
void displayMap(int i,int j){
  //This for loop is responsible for rows
  for(j=1; j<11; j++){
    //This for loop is responsible for columns
    for(i=1; i<11; i++){
      //This if statement will put an 'X' if the "coordinates" matches the ones scanned from the user
      if(i==x && j==y){
        printf("X");
      }
      //This else prints the dashes
      else{
      printf("-");
      }
    }
    //This printf will change the line after the 10 dashes in a row
      printf("\n");
  }
}

//This function will be responsible to move the 'X' around the map
void move(char m){
  //If the user presses 'a' program will decrement 1 to x, making the 'X' move left 1 space
  if(letter=='A'){
    x--;
  }
  //If the user presses 'd' program will increment 1 to x, making the 'X' move right 1 space
  else if(letter=='D'){
    x++;
  }
  //If the user presses 'w' program will decrement 1 to x, making the 'X' move up 1 space
  else if(letter=='W'){
    y--;
  }
  //If the user presses 'a' program will increment 1 to x, making the 'X' move down 1 space
  else if(letter=='S'){
    y++;
  }
  //In case the x variable reaches 0 it will change it's value to 10 in order to make the 'X' appear on the opposite side of the grid
  if(x==0){
    x=10;
  }
  //In case the x variable reaches a a value above 10 , it will give an absolute(positive) value between 1 to 10, the purpose it that when the x reaches 11, we will have 11/10=1, and with that we get the 'X' appear on the opposite side of the grid
  else if(x>10){
    x=abs(x/10);
  }
  //In case the y variable reaches 0 it will change it's value to 10 in order to make the 'X' appear on the opposite side of the grid
  if(y==0){
    y=10;
  }
  //In case the y variable reaches a a value above 10 , it will give an absolute(positive) value between 1 to 10, the purpose it that when the y reaches 11, we will have 11/10=1, and with that we get the 'X' appear on the opposite side of the grid
  else if(y>10){
    y=abs(y/10);
  }
}


int main(void) {


  //Asking the user to choose the first place to have the "X"
  printf("Please enter an X coordinate (a number between 1 and 10):\n");
  scanf("%d", &x);
  printf("Please enter an Y coordinate (a number between 1 and 10):\n");
  scanf("%d", &y);
  while ((ch = getchar()) != '\n' && ch != EOF);
  do{
    //This will display the map in the console
    displayMap(i, j);

  printf("Please enter the letter which corresponds with your choice:\n");
  printf("w - The X will move up 1 space\n");
  printf("a - The X will move left 1 space\n");
  printf("s - The X will move down 1 space\n");
  printf("d - The X will move right 1 space\n");
  printf("x - Exit program\n");
  //This scanf will get the variable responsible to "move" the 'X'
  scanf("%c",&letter);
  //This while loop as the objective to clear input buffer
  while ((ch = getchar()) != '\n' && ch != EOF);

  while(letter!='w' && letter!='a' && letter!='s' && letter!= 'd'  && letter!='W' && letter!='A' && letter!='S' && letter!= 'D' && letter!= 'x' && letter!= 'X'){

    printf("Sorry, your input was not recognised, please enter either w, a, s, d or x:\n");
    scanf("%c", &letter);
    while ((ch = getchar()) != '\n' && ch != EOF);
  }
  //toupper function will transform the letter scanned in uppercase, in this case there is no need to have an or condition in if statements
  letter = toupper(letter);
  move(letter);
  }
  //Thise while loop will make the program run untill the user presses 'x' when on the menu
  while(letter!='x' && letter!='X');
  return 0;
}
```

Console:



Assessment sheet 8

I started this Problem Solving course with just some basic of C from self-learning over the past two years so me and C already met before. My expectations were to learn how to have variables interacting with each other and how to develop coding that could help me to solve programming problems.

Since I already knew some basics, I got familiar when we started talking about variables (integers, doubles, floats, char) even though I thought that it could get my mind in circles, following the online lessons with the examples given by Darren step by step, seeing the doubts of my colleges and my questions answered with live coding examples helped me a lot to know where should I focus my attention when getting variables together, and I would say that something that looks simple but seems to me really important is giving the variable a name that reminds us on what are we going to use that variable for (in the first assessment we used variables mileage and cost) and we can organize our logic really quickly because those names just came naturally when we needed them. One more thing that I think every programmer should do and was asked in every assessment was to explain the code, we could either explain it in the portfolio or comment directly in the code, so, I started having comments at the same time I was coding and in the end, I had a coding file with a explained instructions basically, because I could just take a look at a line of code and instead of reading a full “printf” it would say right above it: “//This line prints the area of the circle”(assessment 3 – menu option B).

Once we got to loops I felt comfortable with it, I had already worked on simple coding with the weekly tasks and the assessment 1 and after doing the weekly tasks with loops and reading the notes, I got it in my head that it was just repeating simple coding lines and that's what a loops does, it just repeats as many times as we want a part of our code and with loops I was able to reduce the size of assessment 2 from maybe 90 lines to 36 and I can see this in a big scale, if we say we reduced 60% of coding lines, in 900 lines we could only have 300 and so on.

Besides what we were taught in class and asked for in the assessments, we had the option to go further and do self-research for a better grade which I thought I could challenge myself and go for it, and I would say it really paid off. At first it can really get confusing with so many online places to search for and so much information that I got confused in the beginning, I started by searching different approaches to if statements which I found switch statements and having two different ways of programming the same thing allowed me to learn different approaches to solving a problem and I was able to see the possibilities for the future where I can use the one that suits the best of my needs.

Then we started arrays. Going from using variables to arrays got a bit confusing because we used different variables to store different values and now, we wanted to store different values in just one variable. On the assessment 5 we worked with arrays and I would say that from my experience this assessment really helped me to better understand arrays because the problem consisted in repopulate the array, display all values, replace one number, and calculate the mean, so these four options made me work in:

- Creating a new array having to give values to each element of the array
- Creating a loop so show me all the elements that I got in the array
- Selecting a specific element on the array and change its value
- Interacting with all the elements in the array, adding them all into one temporary variable and used that to calculate what we want (the mean in this case)

I had a few minutes of brainstorming before figuring it out on how to repopulate the array but ended up deciding that I could just go by going from the first element of the array and defining values for each and in the end the array had been repopulated. In the end of these assessment, I felt confident to work with arrays in the future (like an array studentClass where the students are organized by age) has I can see how having a variable with different values can be useful instead of different values every time.

In the end but not the least, we reached functions which I think it felt right to be in the end and should have all our focus. Until that moment, if we wanted to do a longer program but within the same focus, we for sure would be doing the same loops and the same algorithms all over again and that would not be just time consuming but memory consuming as well, and the functions came to solve that. Creating a function outside the main and having the algorithm ready to use whenever we just by calling it in main it is a time saver, and the code will look much cleaner.

After all the coding I feel more confident to work on harder problems and I can easily read the code and identify where the errors are, which by experience I would say they always are in the smallest places (always look for ';'!) and I think that the better way to improve our weaknesses is practicing and see other persons logic when coding.