REPORT OF PROJECT

I have completed the project so I will mention that what I do in the project with summarizing it.First, for the menu of the program I have used switch statement.I have used global values like selection and x because I have used that variables also in functions.I have benefited from nested structures because I had to show the menu again and again after every selection.For this reason for every program I have created a function and I have called these functions in part of cases.I have used switch statement in while statement and I have inserted respectively function\_program(1 or 2 or 3 or 4 or 5..), function\_selection() (for selection I have created a different function) at the end of the every case.For case 0 I have also created a function to exit from the program and I have also edited case 0 because it had to be.So we can continue with programs.

# Program 1:Solving The Mathematical Equation

In this program first I have defined that variables.I have get the variables N, R, S from memory.Also I did diagnostic check for N, R, S values because this values can’t be negative or 0.For calculations with every single symbols( like add, product ) I have used for statement and in for statement I have used required formulas for every symbol (i.e from 1 to N… formula) .I have also used type casting at necessary points and I have also used sqrt(), pow() functions with math.h library.End of the program I have printed the results with printf.

# Program 2:Drawing The Diamond Shape

In this program I had to use so many variables.So I can have used arrays but I have not use it.I also have used three for statements but this can be developed to two for statements so on and I have prefered this method so I did it.I have did diagnostic check for odd number which can’t be even number or negative number and I have used or gate for this reason.I have assigned variables to another variables because these variables were necessaried to me and I had to control ( odd / 2 + 1 ) and ( odd ) which my median value and line – column value that according to these I stored these value first then after every newline I changed theses value by means of requiring(i.e . c+=2 a--).I have used tricks like x--, x = 0 and y = 0 to fit to if-else conditions these values and make the conditions operate properly.In addition to this you can see the printf part in the program in the for statements and you can recognize that what I did there.

# Program 3:Printing Keyboard Inputs In Reverse

I have to admit that I couldn’t write this program.So I had to learn it and for this reason I gotta look it.In this program c had defined as char.With scanf c character is entered by user.The key statement is;

if(c != '\n')

{

function\_reverse\_sentence();

printf("%c", c);

}

At this point when I enter a character that not equals to '\n' the program goes into if statement and again gets the character from user because the function is called again.

This situation continues till user enters '\n' character.Thus far printf("%c", c); piles up with nested if statements.When user enters '\n', the printf("%c", c); statements get printed from inner if statement to outer if statement and program is terminated.But I will touch on an important problem when I transferring this program to the menu.When I entered my selection the program is not stopping at scanf which in this program.The program is detecting '\n' immediately when I press enter in selection menu.So at this point I have proposed to this problem like this;

if(c != '0')

{

function\_reverse\_sentence();

printf("%c", c);

}

The 0 is that constant value for control the condition.This means that you must enter abcd0 and program will reverse it like that;

dcba

# Program 4:Find The Second Smallest Element

In this program I have created a 50 – sized array and variable size\_t i.You can increase its size in the direction of your prefers.Also I have created a function to find the second smallest element.I got the variables from memory with scanf.I also had diagnostic check in addition to project in the program.Firstly I had to create numbers[ 0 ], after I used while loop and for loop.In first while loop I controlled numbers[ 0 ] till it takes a positive value and also it can’t take -1 sentinel value because it is starting value.In for loop I assigned i = 1 because numbers[ 0 ] is created.The condition is numbers[ i - 1 ] != -1 which is sentinel value(i - 1 because I had to control previous values such that i variable increase by 1 in every repetition).I used if statement and while statement in for loop with nested.If­­-statement controls the positive number and gets the new value for new variable of the array from the user.While-statement controls the value of new variable from the user until the variable has a positive integer.Let’s go to the function.If you pay attention firstly I created values of variables of array then passed the array to function because I controlled that values after created them.

In function I defined i as size\_t and smallest, secondsmallest as integer.First I assigned a[ 0 ] to secondsmallest and smallest.In for loop I controlled every value of array that I created.

if( a[ i ] < smallest ){

secondsmallest = smallest;

smallest = a[ i ];

}

In this condition I did essential assignments.

else if( a[ i ] > smallest && smallest == secondsmallest)

secondsmallest = a[ i ];

This condition is if the a[ 1 ] > smallest for the first control

else if( a[ i ] > smallest && a[ i ] < secondsmallest )

secondsmallest = a[ i ];

And this condition for if a[ i ] is in this range

# Program 5:Distinct Random Number Generation

In this program I defined number[ SIZE ] SIZE = 20 as integer and i, a as size\_t.Numbers must be randomize as temporal so I used srand( time( NULL ) ) function.Firstly I assigned number[ 0 ] = rand() % 20 + 1 so I had created that.After, I used for loop from i = 1 to SIZE that equals to 20.In for loop first I used this equality a = i because a had to be controller because I controlled i with every previous variables values.Again for every repetition I created value for each variable.

a = i;

number[ i ] = rand() % 20 + 1;

while( a > 0){

if( number[ i ] == number[ a - 1] ){

number[ i ] = rand() % 20 + 1;

a = i + 1;

}

a--;

}

You can see that this is the key part of the program.

While a > 0 and until a = 0 this part of the program continues.For instance i = 3 so a = 3 and in while loop if( number[ i ] == number[ a - 1] ) when this equalized the while loop must be start again from i = a in other words i = 3, a = 3 must be.Because I must control i = 3 again with every previous value.And in while loop if( number[ i ] == number[ a - 1] ) this statement mustn’t equalized so a can equals to zero.And my generated value is different from previous values.And while loop terminates.

I must say that this is epicentrum!!!

a = i + 1;

a--;

The rest of the program is the easiest part of the program I must say that.

for( i = 0; i < SIZE; i++ ){

printf( "%d ", number[ i ] );

}

And program executes

İSMAİL BAŞ 150315040 CBÜ CSE 1.ÖĞRETİM

# 