**מעבדה 5 – מת"מ -   
מגישים:  
אורי מלכא – 314862996  
אלן ציפין - 313206062  
  
שאלה 1 קוד:  
שאלה 1 פלט:** **שאלה 2 קוד:  
  
  
  
שאלה 2 פלטים:** **שאלה 3 קוד:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void Error\_Msg(char\* str);

/\*

Function name:Error\_Msg

Input:string

Output:void

Function Algorithm: exits and display string.

\*/

int main()

{

FILE\* in, \* out;

char tempstr[17]; //each student information is stored in a 16 letters string 6for name 4for Section and 6 for 2 grades.

char TargtedSection[5] = "Comp";

if (!(in = fopen("Students.txt", "r")))Error\_Msg("The input file is wrong");

if (!(out = fopen("StudentsNew.txt", "w")))Error\_Msg("The output file is wrong");

while (fgets(tempstr,17,in)!=NULL ) { //we read each time 16 letters

char tempSection[5]; //this is to temporary save the section name

strncpy(tempSection, tempstr + 6, 4); //here we copy the section name of each student

tempSection[4] = '\0';

if (!strcmp(TargtedSection, tempSection)) //we compare if the student is in the targeted section.

{

char tempName[7], tempGrade1[4], tempGrade2[4];

float avg = 0;

//here we take the student name and his 2 grades into a temp strings

strncpy(tempName, tempstr, 6);

tempName[6] = '\0';

strncpy(tempGrade1, tempstr+10, 3);

tempGrade1[3] = '\0';

strncpy(tempGrade2, tempstr+13, 3);

tempGrade2[3] = '\0';

avg = (atoi(tempGrade1) + atoi(tempGrade2)) / 2.0; //we calc his avg

fprintf(out, "%s %.2f\n", tempName,avg); //we write onto the output file.

}

}

fclose(in);

fclose(out);

return 0;

}

void Error\_Msg(char\* str)

{

printf("\n%s", str);

exit(1);

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#define N 32 //for 32 bit int unsigned

int Bit\_Count(unsigned int x);

/\*

Function name: Bit\_Count

Input: unsigned int

Output: int

Function Algorithm: using mask to find the "1" numbers in the binary number representation of x;

\*/

void Bin\_Print(unsigned int x);

/\*

Function name: Bit\_Print

Input: unsigned int

Output: void

Function Algorithm: using mask to print the binary representation of the x

\*/

int main()

{

unsigned int x;

printf("Enter a number:");

scanf("%d", &x);

printf("There are %d bits equal to one in %d\n", Bit\_Count(x), x);

printf("The binary representation of %d is ", x);

Bin\_Print(x);

return 0;

}

int Bit\_Count(unsigned int x)

{

int counter = 0;

unsigned int mask = 1;

while (x != 0) {

unsigned int temp = x;

if (temp & mask) //mask is 000000000.....01 so we sort of doing num%10 to find if its 1 or 0

counter++;

x>>=1;

}

return counter;

}

void Bin\_Print(unsigned int x)

{

int i;

unsigned int mask = 1;

mask <<= N - 1; //the mask is 10............00000

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

int temp = 0;

if (x & mask) //if the result of x\*mask is not 0 its meants that the MSB is 1

temp = 1;

printf("%d", temp);

x <<= 1;

}

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<stdio.h>

#include<stdlib.h>

#define N 8 //for 8bit

void BinPrint(unsigned char ch);

/\*

Function name:BinPrint

Input:char

Output:void

Function Algorithm: prints the binary representation of a number

\*/

unsigned char check\_ms(unsigned char ch);

/\*

Function name:check\_ms

Input:unsigned char

Output:unsigned char

Function Algorithm: checks if the mab is 1

\*/

unsigned char change\_bit(unsigned char ch,int n);

/\*

Function name:change\_bit

Input:unsigned char,int

Output:unsigned char

Function Algorithm: changes the n bit (from the left)

\*/

int main()

{

unsigned char num1 = 102, num2 =102;

int n;

printf("\nThe first part :");

printf("\nThe number is %d ", num1);

BinPrint(num1);

num1 = check\_ms(num1);

printf("\nThe new number is: %d ", num1);

BinPrint(num1);

printf("\n\nThe second part :");

printf("\nThe number is %d ", num2);

BinPrint(num2);

printf("\nEnter a number of the bit to change(1-8):");

scanf("%d", &n);

num2 = change\_bit(num2, n);

printf("\nThe new number is: %d ", num2);

BinPrint(num2); /\*Binary representation of number\*/

return 0;

}

void BinPrint(unsigned char ch)

{

int i;

unsigned char mask = 1;

mask <<= N-1;

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

int temp = 0;

if (ch & mask)

temp = 1; //if the bit in the n place is 1 we print 1

printf("%d", temp);

ch <<= 1;

}

}

unsigned char check\_ms(unsigned char ch) {

unsigned char mask = 1;

mask <<= N - 1; //mask to check msb sign.

if (!(mask & ch)) //if the msb is zero we add 1 to ch.

ch |= mask;

return ch;

}

unsigned char change\_bit(unsigned char ch, int n) {

//11111....N=0.1111 - AND IF ITS 1 ->0

//00000.....N=1.0000 - OR IF ITS 0 ->1

unsigned char OrMask=1, AndMask;

OrMask <<= n;

AndMask = ~OrMask;

if ((ch & OrMask) == OrMask) //this means that the bit in the n place is 1

{

ch &= AndMask; //now the bit in the n place is 0

}

// if failed it means that the bit in the n place is 0

else ch |= OrMask; //now the bit in in the n place is 1

return ch;

}

**שאלה 3 פלטים:**