Final Project - Math Quiz

CompE 271

Grigoli Vashakidze

San Diego State University – Georgia

MATH QUIZ

Math Game in assembly is an interactive math quiz program. It will generate problems in console and user will have to input the answers. Starting point of the program is menu, where user selects to start the game, see the score or exit the quiz.

By choosing “1” to start, the user goes to the first question. Each round consist of four math equations and the user should correctly answer each of them. If the question is answered correctly, one point will be given to the user. Otherwise, user losses one point for every incorrect answer. Scores are accumulative and overall score can be viewed from menu by entering “2”. The user will exit the program by inputting “3”.

The variables of the equations will be generated by pseudorandom number algorithm. There will be four type of equations and as user plays new round he/she have to solve the equation with different numbers. The program will compute answers of these equations using arithmetic operations and will compare to user input.

For demonstration, Project uses Windows system PC with **Code::Blocks.** The project uses x86 32-bit assembly functions written in AT&T syntax.

Time spent on the project 52 hours

**Pseudocode:**

START

**ADD function**

ASSIGN a = random number

ASSIGN b = random number

ASSIGN a + b =sumansw

SCANF U\_IN

IF (U-IN is equal to sumansw)

PRINT Answer correct

Increase score by 1

ELSE

PRINT Answer incorrect

Decrease score by 1

**MULTIPLY function**

ASSIGN a = random number

ASSIGN b = random number

ASSIGN a \* b = sumansw

SCANF U\_IN

IF (U-IN is equal to sumansw)

PRINT Answer correct

Increase score by 1

ELSE

PRINT Answer incorrect

Decrease score by 1

**DIVIDE function**

ASSIGN a = random number

ASSIGN b = random number

ASSIGN a / b =sumansw

SCANF U\_IN

IF (U-IN is equal to sumansw)

PRINT Answer correct

Increase score by 1

ELSE

PRINT Answer incorrect

Decrease score by 1

**SUBTRACT** **function**

ASSIGN a = random number

ASSIGN b = random number

ASSIGN a - b =sumansw

SCANF U\_IN

IF (U-IN is equal to sumansw)

PRINT Answer correct

Increase score by 1

ELSE

PRINT Answer incorrect

Decrease score by 1

**MAIN function**

PRINTF please enter a number 1 for start, 2 for score , 3 for exit

SCAN USER\_IN

IF (USER\_IN is 1)

CALL Assembly function for random number

IF (USER\_IN is 2)

PRINT current score

IF (USER\_IN is 3)

EXIT

ELSE

PRINT input is incorrect

**C function:**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

**extern unsigned** RandomG(); // Import assembly function for random number generator

**extern int** Add(); //Import assembly function for addition

**extern int** Mul();//Import assembly function for multiplication

**extern int** Division();//Import assembly function for division

**extern int** Sub();//Import assembly function for subtraction

**int** n, a,b,in;

**int** answ;

**unsigned** score = 0;

**void** scanandcheck(){ // function to scan user input and check answer right or not

scanf("%d", &in);

**if** (answ == in){

++score;

printf("\nCorrect!!!\t\t\t\t\t\t Score:%u\n\n",score);

}**else**{

**if** (score!=0)--score;

printf("\nIncorrect (1 point lost)!!! The CORRECT Answer is: %d\t\t\t\t\t\t Score:%u\n\n1",answ,score);

}

}

**void** addf(){ // addition function gets two Four-digit randomly generated numbers

a = 1000+(RandomG(time(NULL)))%9000; // Gets randomly generated number and passes time argument for LFSR seed value to generate different numbers

b = 1000+(RandomG())%9000;

answ = Add(a,b);

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\* Question 1: ADDITION \*\n");

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t\t%d + %d = ?\n",a,b);

printf("\t\tPlease enter your answer:");

scanandcheck();

}

**void** divf(){ //divides Four-digit two randomly generated numbers

a = 1000+(RandomG(time(NULL)))%9000; // Gets randomly generated number and passes time argument for LFSR seed value to generate different numbers

b = 1000+(RandomG())%9000;

answ = Division(a,b);

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\* Question 4: Division \*\n");

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t%d / %d = ?\n",a,b);

printf("\tPlease ONLY enter the answer as a quotient integer:");

scanandcheck();

printf("\n\tQuiz Finished! Your overall score is : %d \n",score);

main();

}

**void** subf(){ // Substracts Four-digit randomly generated numbers

a = 1000+(RandomG(time(NULL)))%9000; // Gets randomly generated number and passes time argument for LFSR seed value to generate different numbers

b = 1000+(RandomG())%9000;

answ = Sub(a,b);

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\* Question 2: SUBTRACTION \*\n");

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t%d - %d = ?\n",a,b);

printf("\tPlease enter your answer:");

scanandcheck();

}

**void** mulf(){ // Performs multiplication on Three-digit and Two-digit randomly generated numbers

a = 100+(RandomG(time(NULL)))%900; // Gets randomly generated number and passes time argument for LFSR seed value to generate different numbers

b = 10+(RandomG())%90;

answ = Mul(a,b);

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\* Question 3: Multiplication \*\n");

printf("\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t%d \* %d = ?\n",a,b);

printf("\tPlease enter your answer:");

scanandcheck();

}

**int** main(**void**) {

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\n\t\t\t\tWelcome to the math quiz\n\n\t\t This is a quiz with four questions on simple arithmetics\n\n\tIf you answer question correctly you will get 1 point; Otherwise you loose 1 point\n\n");

printf("Please enter:\n\n\t1 - Start the Quiz\n\t2 - Score\n\t3 - EXIT\n");

scanf("%d",&n);

**if** (n==1){

addf();

subf();

mulf();

divf();

}**else if**(n==2){

printf("\t Your Score is: %d\n",score);

main();

}**else if**(n==3){

**return**;

}**else**{

printf("Invalid input\n");

main();

}

**return** EXIT\_SUCCESS;

}

**Assembly Function:**

.global \_RandomG

.global \_Add

.global \_Mul

.global \_Division

.global \_Sub

\_RandomG:

pushl %ebp /\* save old stack \*/

movl %esp, %ebp /\* new stack \*/

subl $16, %esp /\* allocate 16 bytes \*/

movl $0, 12(%esp) /\* For loop starting from 0 \*/

jmp FK

LFSR:

movl 8(%esp), %ebx /\* Seed value is stored in EBX \*/

shrl %ebx /\* Seed value is shifter right \*/

movl %ebx, %eax /\* moves shifted value to EAX \*/

movl 8(%esp), %ebx /\* EBX gets again Seed value value \*/

andl $1, %ebx /\* Performs logical AND and stores in EBX \*/

negl %ebx /\* Twoâ€™s complement inversion of EBX) \*/

andl $0xA3000000, %ebx /\* Logical AND 0xA3000000 that is for taps 32, 30, 26, 25 \*/

xorl %eax, %ebx /\* Logical XOR EAX shifted value and EBX \*/

movl %ebx, 8(%esp) /\* Moves result value to +8 ESP and changes previous seed value \*/

addl $1, 12(%esp) /\* increases +12 starting value by one \*/

FK:

cmpl $19, 12(%esp) /\* compares +12 value to 19 for the FOR loop , it compares 19 because starts from 0 and iterates 20 times\*/

jbe LFSR /\* Starts again for loop \*/

movl $0, %ebx /\* Sets EBX value to \*/

leave

ret /\* leave the function \*/

\_Add:

/\*Setup stack\*/

pushl %ebp

movl %esp, %ebp

subl $16, %esp

movl 8(%ebp), %eax /\* Get value of a \*/

movl 12(%ebp), %ebx /\* Get value of b \*/

addl %ebx, %eax /\* adds %ebx to %eax and sum is returned by %eax \*/

leave

ret /\* leave the function \*/

\_Mul:

/\*Setup stack\*/

pushl %ebp

movl %esp, %ebp

subl $16, %esp

movl 8(%ebp), %eax /\* Get value of a \*/

movl 12(%ebp), %ebx /\* Get value of b \*/

imull %ebx, %eax /\* perform multiplication \*/

leave

ret /\* leave the function \*/

\_Division:

/\*Setup stack\*/

pushl %ebp

movl %esp, %ebp

subl $16, %esp

xorl %edx, %edx /\* Zero %edx \*/

movl 8(%ebp), %eax /\* Get value of a\*/

movl 12(%ebp), %ebx /\* Get value of b\*/

divl %ebx /\* divides %eax by %ebx and quotient is returned\*/

leave

ret /\* leave the function \*/

\_Sub:

/\*Setup stack\*/

pushl %ebp

movl %esp, %ebp

subl $16, %esp

movl 8(%ebp), %eax /\* Get value of a\*/

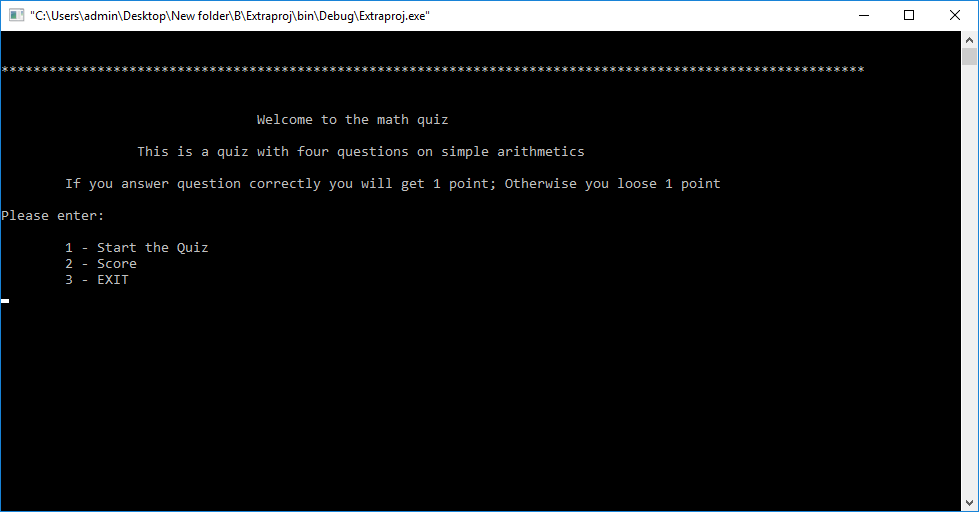
movl 12(%ebp), %ebx /\* Get value of b\*/

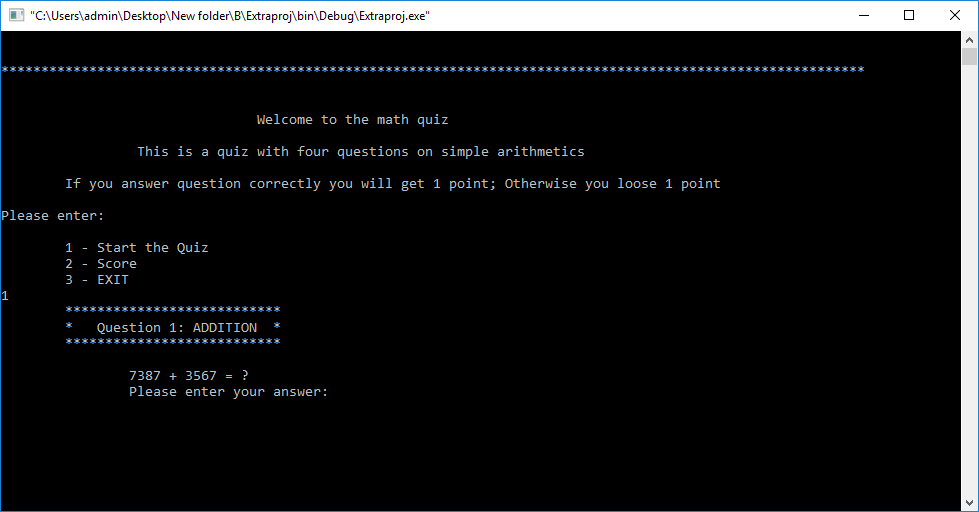
subl %ebx, %eax /\* subtract b from a\*/

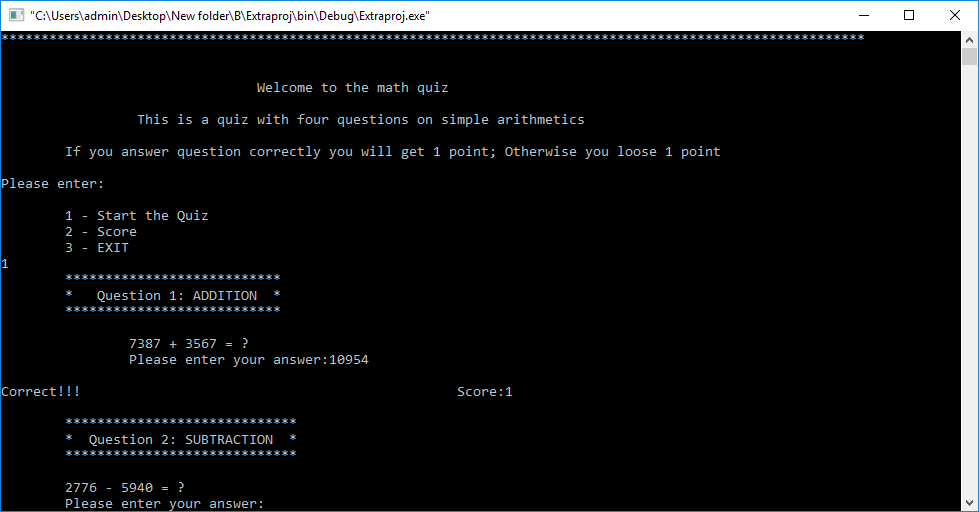
leave

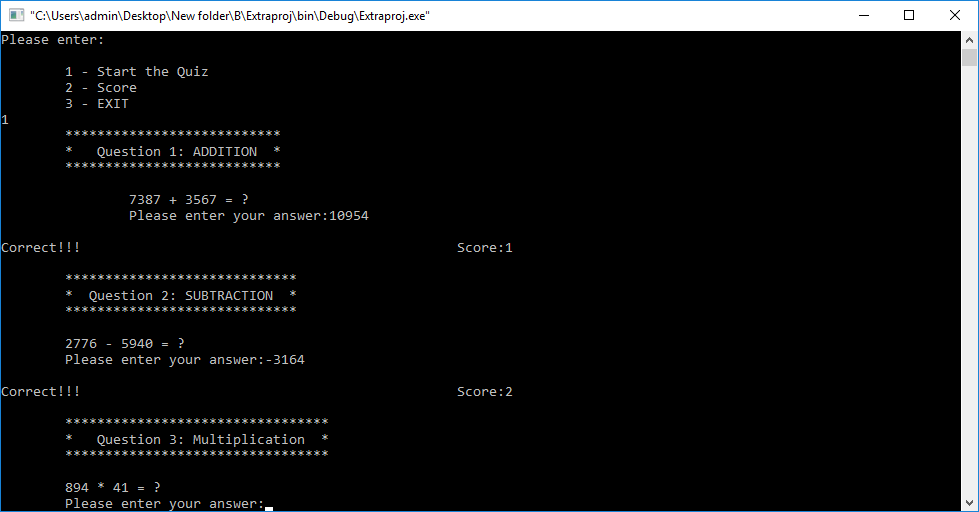
ret /\* leave the function \*/

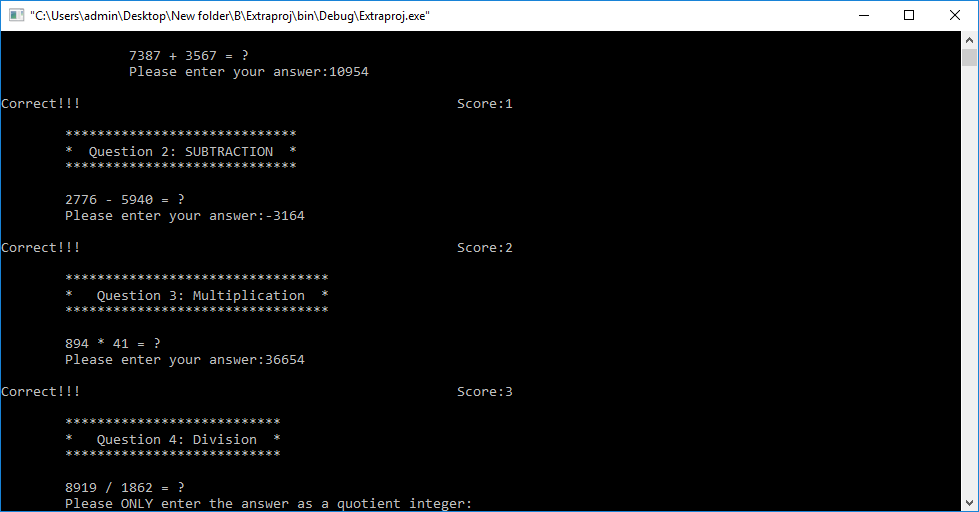
**Screenshots:**

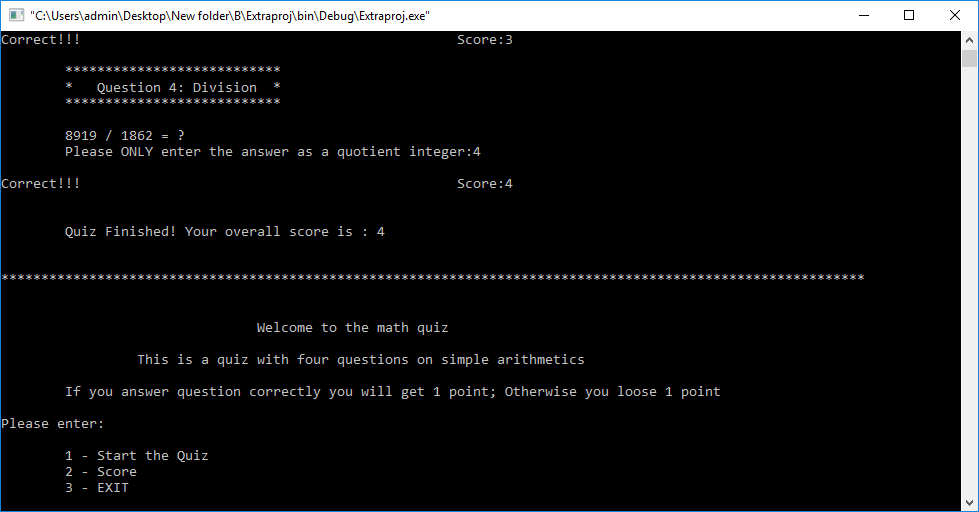
Starting screen - 

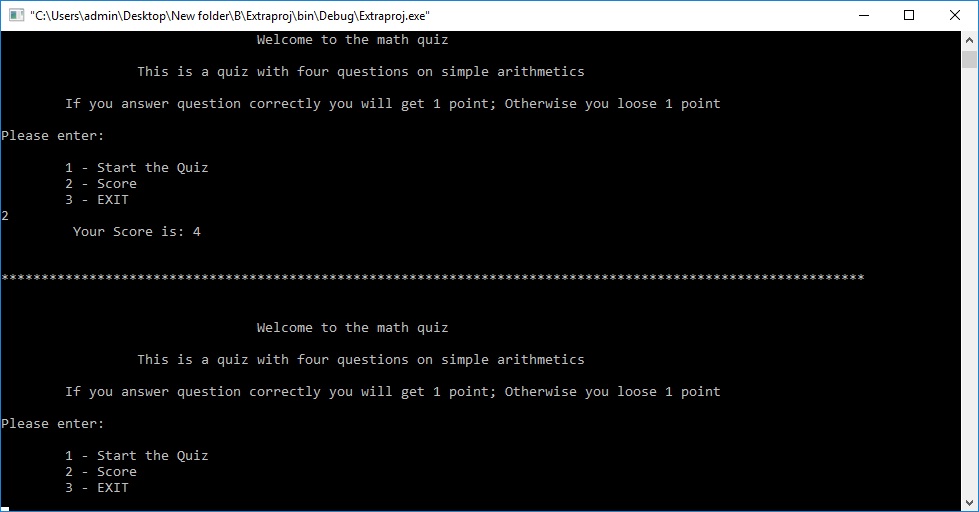
Start the Quiz - 

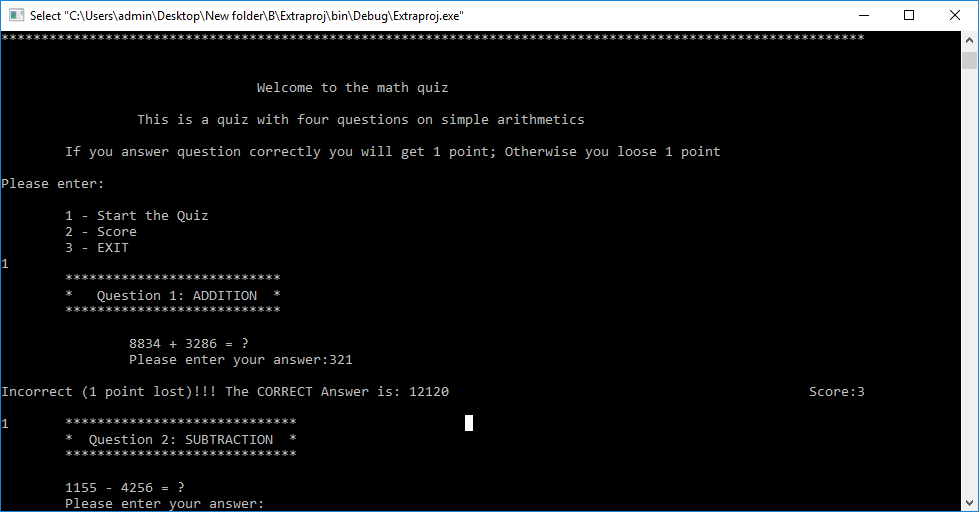
Question 1 answered correctly – 

Question 2 answered correctly - 

Question 3 answered correctly - 

Question 4 answered correctly - 

See the Score - 

Test incorrect answer on question – 

Test incorrect answer - 