Student Name: _____

Student ID: _____

1	2	3	4	5	6	Total

Marmara University – Faculty of Engineering

Computer Engineering

Spring 2019-2020

CSE 2138 – Systems Programming

Online Homework #1

05 June 2020

I hereby swear that the work done on this online homework is totally my own; and on my honor, I have neither given nor received any unauthorized and/or inappropriate assistance for this homework. I understand that by the school code, violation of these principles will lead to a zero grade and is subject to harsh discipline issues.

Full Name: *Signature:*

Solution and Submission Notes:

1. Book, slides and notes are open.
2. Internet usage is not allowed; however, computers and cell phones may be used only for viewing course materials provided for this course.
3. This online homework consists of 6 questions and 7 pages.
4. Please write your answers clearly and neatly to the appropriate place.
5. **Show all your work.**
6. Write your student ID, first name and last name, and total number of pages you used on top of each solution page.
7. If you have not printed this PDF file, write the sentences in the frame above and the date, in your handwriting, on top of the first page of your answer sheet, and sign it.
8. Write your solutions in your handwriting on blank A4 sheets.
9. Scan all the solution pages to a single PDF file, name it as "FirstName_LastName_ID.pdf".
10. Upload your PDF file to http://ues.marmara.edu.tr before deadline.

1. **(16 pts)** Assume that:
   - We are running code on a 10-bit machine using two's complement arithmetic for signed integers
   - `short` integers are encoded using 5 bits
   - Sign extension is performed whenever a `short` is cast to an `int`

   The following definitions are given:

```
int x = -37;
unsigned u = x;
short s = -11;
short sx = (short)x;
```

Fill in the empty boxes in the tables below.

In the table, for each of the C expressions in the first column; either:
- state that it is true for all argument values, or
- give an example where it is not true.

Hint: Be careful with the rules that C uses for signed and unsigned ints.

Show all your work!

	Expression	Decimal Representation i)	Binary Representation ii)
	`0`	0	00 0000 0000
a)	`x`	-37	
b)	`u`		
c)	`s`	-11	
d)	`u / 2`		
e)	`s >> 1`		
f)	`sx`		
g)	`(!u - 1) & u`		
h)	`((int)(u >> 9) + ~0)`		
i)	`((int)u) >> 9`		

Note: If you have not printed this PDF file, write your final answers for corresponding empty cells like:

a.i) a.ii) b.i) …

2. **(16 pts)** Consider two different 8-bit floating point formats based on the IEEE floating point format given below:

Format A:

- There is sign bit in the most significant bit.
- The next 3 bits are the exponent. The bias is $2^{k-1} - 1 = 3$.
- The last 4 bits are the fraction.

Format B:

- There is sign bit in the most significant bit.
- The next 4 bits are the exponent. The bias is $2^{k-1} - 1 = 7$.
- The last 3 bits are the fraction.

For formats A and B, fill in the empty boxes in the table below (use round-to-even when necessary). Express numerical values as fractions (e.g. 183/256)

Show all your work!

		ii)	ii)
	Value	**Format A Bits**	**Format B Bits**
	Zero	0 000 0000	0 0000 000
a)	6		
b)		1 100 1100	
c)			0 0100 100
d)	11/8		

Note: If you have not printed this PDF file, write your final answers for corresponding empty cells like:

a.i)

a.ii)

b.i)

…

3. **(15 pts)** Consider the following C function and its x86-64 assembly code. Fill in the missing parts of the C code to get a program equivalent to the given assembly code.

   Show all your work!

```
int foo(int *A, int m, int n){
  int i;
  int k= _____________;
                a)
  for(i=_______; ___________; i++){
          b)         c)
    if(_______________){
              d)
      k = _____________;
                e)
    }
  }
  return k;
}
```

```
  pushq %rbp
  movq  %rsp, %rbp
  movslq %esi, %rcx
  movl (%rdi,%rcx,4), %eax
  cmpl %edx, %ecx
  jg LBB0_3
  movslq %edx, %rdx
LBB0_2:
  movl 4(%rdi,%rcx,4), %esi
  cmpl %eax, %esi
  cmovgel %esi, %eax
  incq %rcx
  cmpq %rdx, %rcx
  jl LBB0_2
LBB0_3:
  popq  %rbp
  retq
```

4. **(20 pts)** Consider the following C code and the assembly code that implements the branches of the switch statement of C code:

```
long fun(long x, long y){
   long result = 0;
     switch (y) {
       case ________:
              a)
             result = ________;
                          b)
             break;
       case ________:
              c)
             x = __________;
                    d)
       case ________:
              e)
             result = ________;
                          f)
             break;
       case ________:
              g)
       case ________:
              h)
             result = ________;
                          i)
             break;
       default:
             result = ________;
                          j)
     }
     return result;
}
```

```
f5f:   movq  %rdi, %rax
f62:   cmpq  $5, %rsi
f66:   ja    0xf6e
f68:   jmpq *0x5a0(,%rsi,8)
f6a:   addq  %rsi, %rax
f6d:   retq
f6e:   leaq (%rax,%rax,2),%rax
f75:   retq
f76:   shlq  2, %rax
f79:   retq
f7a:   andq  %rsi, %rax
f7d:   decq %rax
f80:   retq
```

The jump table for the switch statement is given below.
You can assume that the first entry in the jump table is for the case when y=0.
Parameters `x` and `y` are passed in registers `%rdi`, and `%rsi`, respectively.

```
0x5a0:  0xf6a
0x5a8:  0xf6e
0x5b0:  0xf76
0x5b8:  0xf6a
0x5c0:  0xf7a
0x5c8:  0xf7d
```

Fill in the blank portions of the C code above to reproduce the function corresponding to the assembly code.

Show all your work!

5. (23 pts) Consider the following C declaration:

```
typedef struct q5 {
      char address[5];
      short code[2];
      char id;
      long data;
      char initial;
      double amount;
      char name[3];
      struct q5 *next;
      int count;
} q5;
```

a) Using the template below (allowing a maximum of 64 bytes), indicate the allocation of data for the `struct q5`. Mark off and label the areas for each individual element. Indicate the parts that are allocated, but not used, with an X. (First byte is marked for you).

`q5:`

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
addr															

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47

48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

b) How many bytes of space in `q5` are allocated for an object of type `q5`?

c) How many bytes of space in `q5` are wasted?

d) Now rewrite the `q5` struct so that the amount of wasted space in `q5` is minimized.

```
typedef struct q5 {




} q5;
```

Show all your work!

6. **(10 pts)** Consider the following C code and the assembly code for the same function:

```
_rec:
  pushq  %rbp
  movq   %rsp, %rbp
  movl   $1, %eax
  cmpq   $rsi, %rdi
  je     LBB0_3
  movq   %rsi, %rcx
  subq   %rdi, %rcx
  movl   $1, %eax
LBB0_2:
  incq   %rax
  addq   %rsi, %rcx
  jne    LBB0_2
LBB0_3:
  popq   %rbp
  retq
```

```
long rec(long x, long y){

  if (________________)
            a)

    return ___________;
              b)
  else{

    long a;
    a = rec(_________, _________);
              c)         d)

    return _______________;
                e)
  }

}
```

Fill in the missing parts of the C code to get a program equivalent to the generated assembly code.

Show all your work!