Name, Surname: ..... **Group : .....**

Student No : ..... **Signature :**.....

Eastern Mediterranean University
Department of Computer Engineering

CMPE 108 – Algorithms and Programming
Sample Final Exam,
Duration: 90 minutes

Q1 15%	
Q2 15%	
Q3 15%	
Q4 15%	
Q5 15%	
Q6 15%	
Q7 10%	
Total 100%	

Instructions:

- There are **7** questions in **6** pages including the cover page.
- Calculators, mobile phones and any electronic devices are **NOT** allowed.
- A table of operators for precedence and associativity is attached on Cover Page.
- Passing any material including rubbers, pencils etc. to anybody else is strictly prohibited in the exam.
- Asking questions to invigilators is prohibited. The instructors will visit the exam rooms regularly.

Precedence and Associativity Table

Operators	Associativity
`( ) [ ] -> .`	Left to right
`! ++ -- + - * & (type)`	Right to left (Unary)
`* / %`	Left to right
`+ -`	Left to right
`< <= > >=`	Left to right
`== !=`	Left to right
`&&`	Left to right
`||`	Left to right
`?:`	Right to left
`= += -= *= /= %=`	Right to left
`,`	Left to right

% specifiers in ANSI C:

`%c` char single character
`%d` (`%i`) int signed integer
`%e` (`%E`) float or double exponential format
`%f` float or double signed decimal
`%g` (`%G`) float or double use `%f` or `%e` as required
`%o` int unsigned octal value
`%p` pointer address stored in pointer
`%s` array of char sequence of characters
`%u` int unsigned decimal
`%x` (`%X`) int unsigned hex value

String Control Codes:

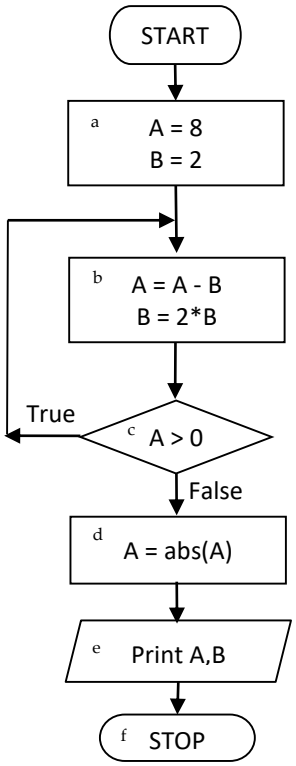
`\b` backspace, `\f` formfeed, `\n` new line, `\r` carriage return, `\t` horizontal tab, `\'` single quote, `\0` null.

Q1 /15p/

Consider the following flowchart, where The function **abs(A)** returns the absolute value of a variable **A** defined in **math.h** header file.

a) Write a C program corresponding to the flowchart by using a do-while loop.

```
#include <stdio.h>
#include <math.h>
int main(void) {
 int A=8, B=2;
……
……
……
……
……
……
……
……
……
……
 return 0;
}
```



b) Trace the flowchart for input value **A=8** and **B=2.**

step	A	B	A>0
a	8	2	

Q2 /15p/.

a. Given the definition of the function

```
int func (int a, int b)
 {
 if( a<b) return a%b;
 else return a/b;
 }
```

and variables, `int x=2, y=3; float z= 1.5;` evaluate the following C expressions individually.

i) `z+func(x, y)` .....

ii) `func(y, x) - y++` .....

iii) `x > y > z` .....

iv) `! (++x != y ) && z` .....

v) `(int) z + --y` .....

vi) `x *= y - 5` .....

b. [2 pts. each] Write down the **C expressions** corresponding to the following arithmetic expressions.

i) $1 + \frac{x+1}{x^2+1}$ .....

ii) $1 + \frac{2}{\frac{1+x}{1/2x}} - 2$ .....

Q3 /15p/

For the following display outputs, fill in the blanks of the printf statements to get the output exacly as given below. Note that in codes and output "_" represents the space character. a.

a. `printf("%.....", 1.2)` to get the output: `_1.20`

b. `printf("%.....",1.2348)` to get the output: `1.235_`

c. `printf("%.....",100.2)` to get the output: `___100.200`

d. `printf("%.....", 10)` to get the output: `_00010`

e. `printf("%.....", 70)` to get the output: `70___`

f. Fill in the blanks of the following `scanf` statement to read a room number into an integer named `roomNo`, and a char named `roomName` correctly. Example input : `108 C` results in reading `roomNo`=`108`, `RoomName`=`C`.

```
int roomNo;
char roomName;

scanf(.....);
```

Q4 /15p/

Consider the code given at the right side in answering to the following parts:

a) What is the output of the following while-loop?

b) Write the corresponding for loop?

code with "while" loop

```c
#include <stdio.h>
int main(void)
{
  int d = 5 ;
  int i = 2 ;
  while( i < 6)
  {
    printf("%i %d\n", d, i);
    d--;
    i++;
  }
  return 0;
}
```

answer box for (a) Output screen /5p/

answer box for (b) "for" loop /10p/

Q5 /15p/

The start of the code is listed in the following part.

```
#include <stdio.h>
#define dim 6
int main(void){
   float prod, x[dim]={1.0, 3.0/2, 2.0, 4.0};
   int i;
```

.**a.** How many numbers can be stored in **x[ ]** safely? .....

.**b.** What are the values of **x[2]** and **x[4]**? **x[2]**= ..... **x[4]**=.....

.**b.** Write a C statement to assign the sum of the first two elements to the last element of **x**.

……

.**c.** Write a for-loop to read new values from the keyboard to the first three elements of the array **x[]**.

……

……

……

……

.**d.** Write a for-loop to find the product of the last three elements of the array **x**. You shall initialize **prod=1**, and in a for loop multiply necessary elements one by one on **prod**.

……

……

……

……

……

……

.**e.** Declare a one-dimensional real number array **y** of the same length and copy all elements of **x** to **y** in reversed order, such that, for array length **4, x[]={1,2,3,4}** shall yield to **y[]={4,3,2,1}.**

……

……

……

……

……

……

……

Q6/15p/

Find the printed sequence of characters on the output screen for the given code.

```
int n = 0;
char ch = 'D';
while (n < 6){
    switch (n){
      case 1:
        printf("%c",ch);
      case 2:
        printf("%c",'U');
        n+=4; ch='B';
        break;
      case 0:
        printf("%c",ch);
        n++;
        ch =ch-3;
        break;
      default:
        printf("!");
        n = 10;
        break;
      case 3:
        printf("%c",ch);
        n = 1000;
        ch = ch - 1;
    }
}
```

Tracing

n	ch	n < 6
……..	……..	……..
……..	……..	……..
……..	……..	……..
……..	……..	……..
……..	……..	……..

Output screen

Q7./10p/

Consider a program that calculates float **y** for the given float values of **x**, **u** and an integer **k**.

The formula to calculate y is : **y(x,u,k)**=$x + \frac{2u-1}{k}$.

The main program reads **x**, **u**, and **k** ; then it writes "**y(**x**,** u**,** k**)** = y(x,u,k)" replacing the underlined variables by their values, such as, if **x**, **u**, **k** are entered "**1.0 4.5 4**" then it shall write "**y(2.0000, 4.5000, 4)= 3.0000** ".

Fill the empty parts of the program specified with dots (.....) to complete the code accordingly.

```
#include <stdio.h>                          /* C program to find  y= x+(2u-1)/k */
float  yxuk(float x, float u, int k) {      /* function y(x,u,v) definition */
  float y =.....;
  return .....;
}
int main(void) {
  float x, u, y;
  int k;
  printf("Enter the values of x, u and integer k \n");
  scanf("%f %f %d", ..... , ..... , .....); /* read values from console*/
  y = .....; /* call the function here  */
  printf("y(%.4f,%.4f,%d) = %..... \n", .....);
  return 0;
}
```