

# ICS 103 Midterm (Term-102)

---

Thursday April 21st, 2011  
10:00 AM – 11:59 AM

<b>NAME</b>	
<b>ID NUMBER</b>	
<b>Section/Instructor</b>	
<b>Lecture time</b>	

<b>Question</b>	<b>Mark scored</b>
Q1 (25)	
Q2 (10)	
Q3 (20)	
Q4 (20)	
Q5 (25)	
<b>Total (100)</b>	

- Students need not to have a calculator for the exam.
- Students should bring pen or pencil and his University photo ID.
- Students will not be allowed a bathroom break. If you need to go, do so before the exam.
- Students are only allowed to ask question in the first 15 minutes of the exam

## Question 1 (25 points = 4 + 3 + 3 + 3 + 3 + 9)

<pre>int j; for(j=1 ; j &lt;= 8 ; j++)     if(j==1    j==4)         printf("*****\n");     else         printf("*   *\n");</pre>	<p>***** 4 marks = perfect A  *   * 3 marks = A with minor offset in the full lines  *   * 2 marks = missing either cross lines of A  ***** 1 mark = printing only solid or hollow</p> <pre>*   * *   * *   * *   *</pre>		
<pre>int a = 4; int b = 7; int c = ++a * b++; printf("%d\t%d\t%d\n", a, b, c);</pre>	<p>5 8 35</p> <p>1 mark for every number(-1 for bad formatting)</p>		
<pre>int i, j; int counter = 0; for (j = 0; j &lt;= 500; j++)     counter++; printf("%d\n", counter); counter = 0; for (j = 100; j &gt; 20; j--)     counter++; printf("%d\n", counter); for (j = 400; j &lt; 500; j++)     counter = j; printf("%d\n", counter);</pre>	<p>501 80 499</p> <p>1 mark for every number. -1 for bad formatting (output in one line for example)</p>		
<pre>int a, b, c, d; int x, y, z; a = 4; b=3; c=10; d=1; x = c/b; y = a * c - d; z = c % 4; printf("%d %d %d", x, y, z);</pre>	<p>3 39 2</p> <p>1 mark for every number. -1 for bad formatting (output in multiple lines). If student made tabs instead of one space, it should be okay.</p>		
<pre>int a, b; double i, j, k; a = 4; b = 10; i = a/b; j = 4.8839; k = 15.522; printf("%f\n%0.3f\n%8.2f", i, j, k);</pre>	<p>0.000000 4.884 ___15.52</p> <p>1 mark for every number. Every answer will be rewarded either 0 or 1. For example if a student does not show the three spaces in the last output, he will get zero for that answer. Any representation of spaces is acceptable. The first output should be double. 0.0 or 0.00 or 0.000 all accepted. We don't care about the number of zeros on the right. The second output has to be in this rounded format or student gets a zero for that part.</p>		
<pre>int i=3, j=1; scanf("%d", &amp;i); do{     for(j=1 ; j&lt;=3 ; j++)         printf("%d ", i * j );     printf("\n");     i--; }while(i&gt;0); printf("%d %d\n", i, j);</pre>	<p><i>If input was 2</i></p> <p>2 4 6 1 2 3 0 4 1pt for each line</p>	<p><i>If input was 1</i></p> <p>1 2 3 0 4 1.5 pt for each line</p>	<p><i>If input was 0</i></p> <p>0 0 0 -1 4 1.5 pt for each line</p>

## Question 2 (10 points = 2 + 2 + 2 + 4)

Convert these mathematical formulas to their equivalent C code. Write your solution in the table below:

1.  $\sin(x^y) - \frac{a-b}{b+\cos(x)}$
2.  $\log(|x - y|)$
3.  $1 - \frac{1}{1+(a-b)^2}$

`sin( pow (x,y) ) - (a-b) / (b + cos(x))`

-1 → missing parentheses around a-b

-1.5 → missing more parentheses. Or using ^ for pow function.

`Log10( abs(x-y) )`

using log, or fabs is also acceptable solution.

0 → if it is an incorrect solution

2 → if the solution is acceptable

`1 - 1 / (1+ pow( (a-b), 2) )`

Using (a-b) \* (a-b) is acceptable

-1 → if there is one missing parenthesis.

-2 → More errors

Write a code fragment (not a complete program) that computes the expression below, where p = 4

$$\sum_{n=1}^{n=20} \frac{1}{n^p}$$

```
double sum; //0.5
int n,p; sum = 0; p=4; //0.5 point

for(n = 1 ; n<=20 ; n++) //1 point
    sum += 1/pow(n,p); //2 point
```

//no deduction for starting from zero (since this example was writing wrong n=0, and was corrected during the exam

### Question 3: (20 points = 4 + 3 + 3 + 10)

Convert the “for” loops to “while” loops.

<pre>for(i=0 ; i &lt; 10 ; i++)     for(j=i ; j &lt;=10; j++)         printf("*");</pre>	<pre>i=0; //0.5 while(i&lt;10) //0.5 {     j=i; //0.5     while(j&lt;=10) //0.5     { printf("*");       j++; //1     }     i++; //1 }</pre>
--	--

Convert the if-statement to “switch”

<pre>if(a == 1)     printf("One"); else if(a == 2)     printf("Two"); else if(a == 3)     printf("Three"); else     printf("Unknown!");</pre>	<pre>switch(a){ //0.5 case 1 :printf("One");break; //0.5 case 2 :printf("Two");break; //0.5 case 3 :printf("Three");break; //0.5 default : printf("Unkown"); //1 }</pre> <p><b>//-1 if student forget break</b></p>
<pre>if(letter == 'A'    letter == 'a')     printf("Excellect");  if(letter == 'B'    letter == 'b')     printf("Very Good");  if(letter == 'C'    letter == 'c')     printf("Good");</pre>	<pre>switch(letter) {     case 'A':     case 'a': printf("Excellent");               break;      case 'B':     case 'b': printf("Very Good");               break;      case 'C':     case 'c': printf("Good");               break; } //one point for each case. -1 for no "break". -2 for using    in the case condition. -1 for NOT including single quotation.</pre>

Write a code-fragment (not a complete program) that prints the numbers from 1 to 1000 to a text file named “data.txt”.

```
FILE * outFile; //1
int i;
outFile = fopen("data.txt","w"); //2

for(i=1 ; i<=1000 ; i++) //3
{
    fprintf(outFile , "%d ", i); //2
    //-1 for not having a space or new line separator
}
fclose(outFile); //2
```

## Question 4: (20 points)

Write a C program that finds the difference between two times given in hours and minutes and outputs the difference in hours and minutes. The format is 24-hours (no AM-PM)

For example if the user enters two times as "14:55" and "16:03" the output would be "1 hour(s) and 8 minute(s)". We will assume that both times are in the same day.

**HINT: Make your program must ask the user to enter four numbers to represent the hours and minutes for both the first & second time. You may convert the user's input to minutes only to perform the operation, but the output must be in hours & minutes format**

//Student might use different method. But this one follows the hints the grading here is the default method. If student does not use the hint, then you can grade it as you see appropriate

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h> //not needed
int main(void)
{
    int h1,m1,h2,m2; //+4 declaring four variables for the number
    int total1, total2;
    int diff;
    //no need for input validation
    printf("Enter the first time hours then minutes : ");
    scanf("%d %d", &h1, &m1);
    //it is okay if student separate each input
    printf("Enter hours for second time : ");
    scanf("%d", &h2);

    printf("Enter minutes for second time : ");
    scanf("%d", &m2);
    //+2 correctly using scanf. There is no need to adhere to a specific format.
    //convert inputs to total minutes +5 for using hint
    total1 = h1 * 60 + m1;
    total2 = h2 * 60 + m2;

    diff = abs(total1 - total2); //absolute difference +5;
    printf("Difference %d hour(s) and %d minute(s)\n", diff/60 , diff % 60);
    // +4 retrieving minutes and hours from total minutes

    system("pause");
    return 0;
}
```

## Question 5: (25 points)

Write a complete C language program that searches for a specific student ID in a text file. Your program should open the file "student.dat" in order to read data. The program should display a message asking for a students ID, and then read the ID. Your program should continue searching in the file until either one of two conditions is met: either the requested student ID is found or the end of file has been reached. When the search finishes, the program should display a message saying that the requested student ID has been found (or that it has not been found).The code should also take care of cases such as a file does not exist.

```
#include<stdio.h>
#include<stdlib.h>

int main(void)
{
    int userInput, id; //+2 declaration of all variable
    int idFound = 0; //assume id was NOT found
    FILE *in; //1pt
    in = fopen("student.dat","r"); //2pt

    printf("Enter an ID : ");//1pt use scanf with prompt
    scanf("%d",&userInput);

    if(in == NULL) //2pt file validation
    {
        printf("file not found!\n");
        return 1; //or exit 1;
    }

    while( fscanf(in,"%d",&id) != EOF) //+4 loop header
    {
        if(id == userInput) //2pt checking
        {
            idFound = 1; //3 pt using a flag properly
            break; //2 pt using a break to end the loop
        }
    }
    if(idFound) //+4 correctly printing when found
        printf("The user input %d was found!\n",userInput);
    else
        printf("The user input %d was NOT found!\n",userInput);

    fclose(in); //2 closing the read stream
    return 0;
}
```