**King Fahd University of Petroleum and Minerals**

**Information and Computer Science Department**

ICS 103: Computer Programming in C

**Summer Semester 2009-2010 (Term-093)**

##### Major Exam-I

**Time: 120 minutes Wednesday, July 21, 2010**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name:** | KEY | | | | | | | | | | |
| **ID#:** |  |  |  |  |  |  |  |  |  |  |

PLEASE CIRCLE YOUR SECTION BELOW:

|  |  |  |  |
| --- | --- | --- | --- |
| Section | 01 | 02 | 03 |
| Instructor | Mr. AHMAD IRFAN | Dr. FARAG AZZEDIN | Dr. AIMAN EL-MALEH |
| Time | SUMT  9:20-10:10am | SUMT  10:30-11:20am | SUMT  10:30-11:20am |

|  |  |  |
| --- | --- | --- |
| Question # | Maximum Marks | Obtained Marks |
| 1 | 12 |  |
| 2 | 8 |  |
| 3 | 10 |  |
| 4 | 10 |  |
| 5 | 8 |  |
| 6 | 9 |  |
| 7 | 8 |  |
| 8 | 8 |  |
| 9 | 13 |  |
| 10 | 14 |  |
| Total | 100 |  |

**Notes.** 1. Make sure you have **Eleven** pages including the cover page.

2. Closed book and notes

3. Write clearly, briefly and precisely

4. Cheating will result in ZERO grade

## Question 1: (12 points)

Fill the blank in each of the following questions:

1. A compiler translates C programs to machine language.
2. Two most common preprocessor directives are #include and #define.
3. Software development includes the following steps:

## specify problem requirements, analyze the problem, design the algorithm to solve the problem, implement the algorithm, test and verify the completed program and maintain and update the program.

1. The linker program combines the output of compiler with various library functions to produce an executable file.
2. The expression (double)(10/4) evaluates to 2.0.
3. The expression 10/(int)2.5 evaluates to 5.
4. The expression 8<=12-4 evaluates to 1.
5. The expression 21/6/2.0 evaluates to 1.5.
6. The expression 12-10%10 evaluates to 12.
7. The expression 1||!0&&0 evaluates to 1.

**Question 2 (8 points)**

Determine all the errors in the following program with clear explanation of the cause of the error:

#include <stdio.h>

#define PI**=**3.14 This should be #define PI 3.14

**void** main(void) This should be int main (void)

{

int i=5;

double c, r;

scanf(“**%f%f**”, **c**, **r**); This should be scanf(“%lf%lf”, &c, &r);

**PI = 3.14159**; Constant cannot be assigned a value

**c = c % (PI \* r)**

This should be c = (int) c % (int)(PI \* r);

as % operation can be only performed on integers.

There is also a missing semicolon at the end of the line

printf(“i=%d c=**%d** \n”, i, c);

This should be printf(“i=%d c=%f \n”, i, c);

return 0;

}

**Question 3 (10 points)**

Consider the following program. What will be the output for the different values of x typed by the user.

|  |  |
| --- | --- |
| Value of x typed  by user | Program output |
| **1** | **3** |
| **2** | **1** |
| **3** | **13** |
| **5** | **8** |
| **6** | **9** |

#include <stdio.h>

int main()

{

int x;

printf("Enter a value for x: ");

scanf("%d", &x);

switch(x) {

case 1: x=x+2;

break;

case 3: x=x+1;

case 5: if(x==4)

x=x+6;

case 6: x=x+3;

break;

default : x=x-1;

}

printf("%d\n",x);

return 0;

}

**Question 4 (10 points)**

Consider the following program. What will be the output for the different values of x typed by the user.

#include <stdio.h>

|  |  |
| --- | --- |
| Value of x typed  by user | Program output |
| **3** | **F** |
| **7** | **B** |
| **20** | **C** |
| **10** | **D** |
| **9** | **A** |

int main() {

int x;

printf("Enter a value for x >");

scanf("%d",&x);

if(x >= 5 ) {

if(x < 10){

if(x > 8)

printf("A");

else

printf("B");

}

else{

if ( x >= 20)

printf("C");

else

printf("D");

}

}

else

printf("F");

return 0;

}

**Question 5 (8 points)**

Show the output of the following program in the space provided below it. Each square corresponds to one space.

#include <stdio.h>

int main(void) {

double i=-19.963;

printf("%6.0f%8.1f\n",i,i);

printf("%3.2f%5d", i,(int)i);

return 0;

}

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **-** | **2** | **0** |  |  |  | **-** | **2** | **0** | **.** | **0** |  |  |  |  |  |
| **-** | **1** | **9** | **.** | **9** | **6** |  |  | **-** | **1** | **9** |  |  |  |  |  |  |  |  |

**Question 6 (9 points)**

**Write the corresponding mathematical or C expression. Assume that all variables are of type double.**

|  |  |
| --- | --- |
| **C EXPRESSION** | **Mathematical Expression** |
| **2.0/3\*(x\*x-sqrt(3\*x))/(pow(x,y)+1)** |  |
| **sqrt(fabs(2-x)+exp(x))/x\*y+1** |  |
| **pow(fabs(x)-x\*y/(x+2),1.0/3)** |  |

**Question 7 (8 points)**

**Write the equivalent C condition so that each of the following statements is satisfied.**

|  |  |
| --- | --- |
| Statement | Equivalent C condition |
| x is not greater than 100 and y is not even | **x<=100 && y%2** |
| x is outside the interval [-5,5] | **x<-5 || x>5** |
| x is a lower case alphabetical character | **x>=’a’ && x<=’z’** |
| x and y are less than or equal 50 | **x<=50 && y<=50** |

**Question 8 (8 points)**

**Assuming that x is declared as int, rewrite the following if-else-if statement using switch statement:**

if(x>=1 && x<=4)

printf(“Interval 1\n”);

else if (x>=5 && x<=7)

printf (“Interval 2\n”);

else if (x==8 || x==9)

printf(“Interval 3\n”);

else

printf(“Outside the correct range\n”);

**switch (x){**

**case 1:**

**case 2:**

**case 3:**

**case 4: printf("Interval 1\n"); break;**

**case 5:**

**case 6:**

**case 7: printf ("Interval 2\n"); break;**

**case 8:**

**case 9: printf ("Interval 3\n"); break;**

**default: printf("Outside the correct range\n");**

**}**

**Question 9 (13 points)**

Write a complete C program that prompts the user for the Cartesian coordinates of two points (x1,y1) and (x2,y2) and displays the entered two points along with their distance computed by the following formula and rounded to two decimal places:

Use appropriate data types for declared variables. Your program should strictly follow the format in the sample execution given below:



**#include <stdio.h>**

**#include <stdlib.h>**

**#include <math.h>**

**int main (void){**

**double x1, y1, x2, y2, dist;**

**printf("Enter the Cartesian coordinates of the first point: ");**

**scanf("%lf%lf", &x1, &y1);**

**printf("Enter the Cartesian coordinates of the second point: ");**

**scanf("%lf%lf", &x2, &y2);**

**dist = sqrt(pow(x2-x1,2)+pow(y2-y1,2));**

**printf("The distance between the two points (%.2f,%.2f) and (%.2f,%.2f) is %.2f\n",x1,y1,x2,y2,dist);**

**system ("pause");**

**return 0;**

**}**

**Question 10 (14 points)**

# The National Earthquake Information Center has asked you to write a program implementing the following decision table to characterize an earthquake based on its Richter scale number. Write a complete C program that prompts the user to enter the Richter scale number and then displays the earthquake characterization. Write the minimum number of conditions needed in your program.

|  |  |
| --- | --- |
| Richter Scale Number (n) | Characterization |
| n < 5.0 | Little or no damage |
| 5.0 ≤ n < 5.5 | Some damage |
| 5.5 ≤ n < 6.5 | Serious damage: walls may crack or fall |
| 6.5 ≤ n < 7.5 | Disaster: houses and buildings may collapse |
| 7.5 ≤ n | Catastrophe: most buildings destroyed |

# Sample executions of the program are given below:





**#include <stdio.h>**

**#include <stdlib.h>**

**int main (void){**

**double n;**

**printf("Enter the Richter Scale Number (n): ");**

**scanf("%lf",&n);**

**if (n < 5.0)**

**printf("Little or no damage\n");**

**else if (n < 5.5)**

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

**else if (n < 6.5)**

**printf("Serious damage: walls may crack or fall\n");**

**else if (n < 7.5)**

**printf("Disaster: houses and buildings may collapse\n");**

**else**

**printf("Catastrophe: most buildings destroyed\n");**

**system ("pause");**

**return 0;**

**}**