**Information and Computer Science Department**

**ICS 103: Computer Programming in C**

**Midterm Exam, Term 171**

**Saturday, November 11, 2017, at 10 AM**

**Duration: 120 minutes**

**SOLUTION KEY**

**Instructor and Section: Select one**

|  |  |
| --- | --- |
| **Instructor** | **Section** |
| Dr. Tareq El-Bassuny  |  [ ] **01** (UT 7 – 7:50) [ ] **02** (UT 8 – 8:50) [ ] **03** (UT 9 – 9:50) |
| Dr. Mohamed Balah |  [ ] **04** (UT 10 – 10:50) [ ] **05** (UT 11 – 11:50) [ ] **21** (UT 13 – 13:50) |
| Dr. Md Rafiul Hassan |  [ ] **06** (UT 12 – 12:50) [ ] **07** (UT 13 – 13:50) |
| Mr. Muhammad Aslam |  [ ] **08** (MW 7 – 7:50) |
| Dr. El-Sayed El-Alfy |  [ ] **09** (MW 8 – 8:50) [ ] **10** (MW 9 – 9:50) |
| Dr. Muhamed Mudawar |  [ ] **11** (MW 10 – 10:50) [ ] **12** (MW 11 – 11:50) |
| Dr. Ali Al-Suwaiyan |  [ ] **13** (MW 12 – 12:50) [ ] **19** (UT 11 – 11:50) [ ] **20** (UT 12 – 12:50) |
| Dr. Ayman Hroub |  [ ] **14** (MW 13 – 13:50) |
| Dr. Uthman Baroudi |  [ ] **15** (UT 7 – 7:50) [ ] **16** (UT 8 – 8:50) |
| Dr. Louai Al-Awami |  [ ] **17** (UT 9 – 9:50) [ ] **18** (UT 10 – 10:50) |
| Dr. Kamal C |  [ ] **22** (MW 7 – 7:50) [ ] **23** (MW 8 – 8:50) [ ] **25** (MW 10 – 10:50) |
| Dr. Samer Arafat |  [ ] **24** (MW 9 – 9:50) [ ] **26** (MW 11 – 11:50) |

**Instructions**:

1. Answer all questions. Make sure your answers are **clear** and **readable**.
2. The exam is closed book and closed notes. No calculators or any helping aides are allowed. Make sure to turn off your mobile phone and keep it in your pocket.

3. If there is no space on the front of the page, use the back of the page. Indicate this clearly.

|  |  |  |  |
| --- | --- | --- | --- |
| **Question** | **Maximum Points** | **Earned Points** | **Remarks** |
| **1** | **10** |  |  |
| **2** | **16** |  |  |
| **3** | **17** |  |  |
| **4** | **22** |  |  |
| **5** | **15** |  |  |
| **6** | **20** |  |  |
| **Total** | **100** |  |  |

**Question 1**

**Fill-in the blanks (5 points)**

1. A **Linker** converts an object file to an executable file.
2. The **CPU (Central Processing Unit)** is the hardware that performs arithmetic and logical operations.
3. In the **Design** stage of software development, pseudo-code or a flowchart is used to show a list of steps to solve a problem.
4. An example of a high-level computer language is **C (or Java, etc)**.
5. Every C program must include one and only one **main** function.

**Multiple Choice: Circle the correct answer (5 points)**

1. Which of the following assignments produces **zero** result?
	1. **result = 9%3 – 1;**
	2. **result = 8%3 – 1;**
	3. **result = 2 – 5%2;**
	4. **result = 2 – 8%3;**
2. **(3 – 4%3 < 5 && 6/4 > 3)** evaluates to:
3. 0
4. 1
5. 2
6. Syntax error in the above expression
7. If an expression does not contain any parentheses, which of the following operators is performed first in the expression?
8. Arithmetic
9. Relational
10. Logical
11. You can’t tell without seeing the expression
12. Which of the following **if**-conditions evaluates to **true or false** depending on the value of average?
13. **if (average > 70.5 && average < 80.5)**
14. **if (average < 70.5 && average > 80.5)**
15. **if (average > 70.5 || average < 80.5)**
16. **if (70.5 < average < 80.5)**
17. Which of the following conditions evaluates to true when the **letter** variable contains **'Z'** in either uppercase or lowercase?
18. **if (letter = 'Z' || letter = 'z')**
19. **if (letter == 'Z' || letter == 'z')**
20. **if (letter = 'Z' && letter = 'z')**
21. **if (letter == 'Z' && letter == 'z')**

**Question 2: Evaluate each of the following expressions (8 points)**

|  |  |
| --- | --- |
| **Expression** | **Value** |
| **10 > 5 > 4** | **0** |
| **4 > 5 || !2 + 1** | **1** |
| **5 < 10 && 0 >= 1** | **0** |
| **5 > 2 \* 2 + 3 == 3** | **0** |
| **8 + 7 / 2 \* 2** | **14** |
| **5 + (double) 7 / 2** | **8.5** |
| **1 + (int) 4.5 \* 2** | **9** |
| **(x && !x) + (3 < 4)** | **1** |

**Use DeMorgan's theorem to complement each logic expression (2, 3, 3 points)**

The **!** operator must NOT appear outside any parentheses

|  |  |
| --- | --- |
| **Logic Expression** | **Complement of Logic Expression** |
| **(n >= 0 && x < y)** | **(n < 0 || x >= y)** |
| **(a > b && (c=='n' || c=='N'))** | **(a<=b || (c!='n' && c!='N'))** |
| **(w<=5 || (x && (y!=60 || y>z)))** | **(w>5 && (!x || (y==60 && y<=z)))** |

**Question 3**

**Implement the following flowchart using nested if-statements. Do NOT make any changes to the flowchart. Do NOT write a full program. Just write the nested if-statements.** **(7 points)**

**x >= 0**

**y < 10**

**x = x + 10**

**y = y \* 10**

**z = x – y**

**x = y**

*true*

*true*

*false*

**x < y**

**z = x \* y**

*true*

*false*

**z = x / y**

*false*

**if (x >= 0) {**

 **if (y < 10) {**

 **z = x – y;**

 **x = y;**

 **}**

**}**

**else {**

 **x = x + 10;**

 **y = y \* 10;**

 **if (x < y)**

 **z = x \* y;**

 **else**

 **z = x / y;**

**}**

**Convert each description into an equivalent C expression (10 points)**

|  |  |
| --- | --- |
| **English Description** | **Equivalent expression in the C language** |
| $$\frac{1}{yz}+\frac{2}{x}$$ | **1/(y\*z) + 2/x** |
| *x* is equal to 3 or 4 | **(x==3 || x==4)** |
| *x* is a lowercase letter or **'M'** | **((x>='a' && x<='z') || x=='M')** |
| *x* is a positive integer less than 100 | **(x>0 && x<100)** |
| *x* and *y* are even integers | **(x%2 == 0 && y%2 == 0)** |

**Question 4: Show the output of the following code fragments**

|  |  |
| --- | --- |
| **Code fragment** | **Output (3 points)** |
| **int n=297, num;****while (n>0) {** **num=n%10;** **n=n/10;** **printf("%d\n",num);****}** | **7****9****2** |
| **Code fragment** | **Output (3 points)** |
| **int x;****printf("Enter x: ");****scanf("%d",&x);****if (x > 6) {** **if (x < 10) {** **if (x >= 8) printf("A");** **else printf("B");** **}** **else {** **if (x >= 20) printf("C");** **else printf("D");** **}****}****else printf("E");** |

|  |  |
| --- | --- |
| **Input** | **Output** |
| **1** | **E** |
| **8** | **A** |
| **15** | **D** |

 |
| **Code fragment** | **Output (3 points)** |
| **int m=6, n, j=0;****for (n=3; m==6; n++) {** **m--;** **while (j != 1) j++;****}****printf("m=%d,n=%d,j=%d",m,n,j);** | **m=5,n=4,j=1** |

|  |  |
| --- | --- |
| **Code fragment** | **Output (4 points)** |
| **int x = 5, y = 2;****do {** **for (y = x; y < 8; y += 2)** **print("%d, %d\n", x, x+y);** **++x;****} while(x < 8);** | **5, 10****5, 12****6, 12****7, 14** |
| **Code fragment** | **Output (6 points)** |
| **int i;****scanf("%d", &i);****switch (i) {** **case 1: i=i+1;** **case 5:** **case 3: i=i/2;** **case 8: break;** **case 2: i=i\*2;**  **default: i=i+1;****}****printf("%d\n", i);** |

|  |  |
| --- | --- |
| **Input** | **Output** |
| **1** | **1** |
| **2** | **5** |
| **3** | **1** |
| **4** | **5** |
| **5** | **2** |
| **8** | **8** |

 |

**/\* 3 points \*/**

**double x = -21.501;**

**printf("%8.2f\n%7.1f\n%5.0f.0", x, x, x);**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **-** | **2** | **1** | **.** | **5** | **0** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **-** | **2** | **1** | **.** | **5** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | **-** | **2** | **2** | **.** | **0** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Question 5: (15 points)**

Write a C program that prompts for and reads the volume **V** of a cylinder [in cubic cm] and the height **h** [in cm]. It then calculates and displays the surface area of the cylinder in cm2. The surface area must be displayed with an appropriate message and in two decimal places.

Your program must use a named constant **PI** with a value of 3.14159

Your program must check the user input **V** and **h**. It must reject any input value less than or equal to zero. The user is given only one chance to re-enter **V** or **h**. The program must terminate with a proper error message if the user re-enters another invalid value for **V** or **h**. Here is a sample run:

|  |  |
| --- | --- |
| Cylinderr2.jpg | $V= πr^{2}h$ $S=2πrh+2πr^{2}$  |

**Enter volume[cm3] and height[cm] of a cylinder: -5.2 1.3**

**Invalid input! Re-enter volume[cm3] and height[cm]: 5.2 1.3**

**Surface Area = 17.22 [cm2]**

**#include <stdio.h>**

**#include <math.h>**

**#define PI 3.14159**

**int main() {**

 **double volume, height, radius, surface\_area;**

 **printf("Enter volume [cm3] and height [cm] of a cylinder: ");**

 **scanf("%lf%lf", &volume, &height);**

 **if (volume <= 0 || height <= 0) {**

 **printf("Invalid input! Re-enter Volume [cm3] and height [cm]: ");**

 **scanf("%lf%lf", &volume, &height);**

 **if (volume <= 0 || height <= 0) {**

 **printf("Invalid input! program terminated.\n");**

 **return -1;**

 **}**

 **}**

 **radius = sqrt(volume / (PI \* height));**

 **surface\_area = 2 \* PI \* radius \* (height + radius);**

 **printf("Surface Area = %.2f [cm2]\n", surface\_area);**

 **return 0;**

**}**

**Question 6: (20 points)**

The binomial coefficient is defined as: $\left(\genfrac{}{}{0pt}{}{n}{k}\right)=\frac{n!}{k!\left(n-k\right)!}$ , where $n\geq k\geq 0$

$n!=n×\left(n-1\right)×\left(n-2\right)× \cdots ×1$, is the factorial of $n$, and $0!=1!=1$.

Write a C program that reads the values of *n* and *k* from the user, computes and displays the binomial coefficient$.$ If the entered value of *n* or *k* is negative, or *n* < *k* then display an error message and terminate the program. Here are sample runs:

**Enter values of n and k: 5 2**

**Binomial coefficient (5, 2) = 10**

**Enter values of n and k: 4 5**

**Error: n cannot be less than k**

**// Solution 1: Using three loops**

**#include <stdio.h>**

**int main() {**

 **int n, k, coefficient, i;**

 **int n\_factorial = 1, k\_factorial = 1, n\_k\_factorial = 1;**

 **printf("Enter values of n and k: ");**

 **scanf("%d%d", &n, &k);**

 **if (n < 0 || k < 0) {**

 **printf("Error: n or k cannot be negative");**

 **return -1;**

 **}**

 **if (n < k) {**

 **printf("Error: n cannot be less than k");**

 **return -1;**

 **}**

 **for(i = n; i > 1; i--) n\_factorial \*= i;**

 **for(i = k; i > 1; i--) k\_factorial \*= i;**

 **for(i = n-k; i > 1; i--) n\_k\_factorial \*= i;**

 **coefficient = n\_factorial / (k\_factorial \* n\_k\_factorial);**

 **printf("Binomial Coefficient (%d, %d) = %d\n", n, k, coefficient);**

 **return 0;**

**}**

**// Solution 2: Using two loops**

**#include <stdio.h>**

**int main() {**

 **int n, k, coefficient, i;**

 **int product1 = 1, product2 = 1;**

 **printf("Enter values of n and k: ");**

 **scanf("%d%d", &n, &k);**

 **if (n < 0 || k < 0) {**

 **printf("Error: n or k cannot be negative");**

 **return -1;**

 **}**

 **if (n < k) {**

 **printf("Error: n cannot be less than k");**

 **return -1;**

 **}**

 **// product1 = n \* (n-1) \* ... \* (k+1)**

 **for(i = n; i > k; i--) product1 \*= i;**

 **// product2 = (n-k) \* (n-k-1)\* ... \* 1**

 **for(i = n-k; i > 1; i--) product2 \*= i;**

 **coefficient = product1 / product2;**

 **printf("Binomial Coefficient (%d, %d) = %d\n", n, k, coefficient);**

 **return 0;**

**}**