King Saud University

College of Computer and Information Sciences
Computer Science Department

Course Code	**CSC 215**
Course Title	**Procedural Programming**
Section No.	
Semester	**Spring 2022 (432)**
Exam	**Midterm Exam I**

Date	**20/03/2022**	**Duration**	**90 minutes**
Student Name			
Student ID			

		Relevant question	**Full mark**	**Student mark**
CLO 1	a) Apply knowledge of computing and mathematics appropriate to the discipline;	1	10	
CLO 2	b) Analyze a problem, and identify and define the computing requirements appropriate to its solution	2	5	
		3	5	
CLO 3	c) Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs.	4	5	
CLO 4	i) Use current techniques, skills, and tools necessary for computing practice.	-	=	

Feedback/Comments:

Question 1: Copy your answer for each of the following questions to the table:

1	2	3	4	5	6	7	8	9	10

1. What is the value of the expression: `3 / 4 && 4 / 3` ?
   - **A.** `0`
   - **B.** `1`
   - **C.** 12
   - **D.** Compilation error

---

2. Given `int a=8,b=2,c=4;` what is the value of the expression: `a + b / c * a - c / b`?
   - **A.** `10.0`
   - **B.** `4`
   - **C.** 6
   - **D.** Compilation error

---

3. For which positive integer values of `x` is the following condition true? `(2 % (x + 1)) > 0`

   - **A.** true for all even values of `x`

   - **B.** true for all odd values of `x`

   - **C.** true for all positive values of `x`

   - **D.** true when `x` is greater than 1

---

4. What is the output of the following code fragment?
   ```
   char string[] = "CSC215";
   char *ptr = string;
   *ptr = *ptr + 2;
   ptr = ptr + 2;
   printf("%c", *ptr);
   printf("%c", *ptr);
   ptr = string;
   printf("%c", *ptr);
   ```
   - **A.** `CSC`
   - **B.** `CCE`
   - **C.** `C215`
   - **D.** Compilation error

---

5. When an array parameter is passed to a function ...

   - **A.** elements of the actual array parameter are copied into elements of the formal array parameter

   - **B.** elements of the formal array parameter are copied into elements of the actual array parameter

   - **C.** formal parameter is a pointer that holds the address of the actual array parameter

   - **D.** programmer must write code which allocates enough space for the function to store the array

---

6. What is the output of the following code fragment?
   ```
   char string[8] = "abcdefg";
   printf ("%s\n", string + 3);
   ```
   - **A.** `abcdefg`
   - **B.** `abc`
   - **C.** `defg`
   - **D.** Compilation Error

---

7. What is the output of the following code fragment?
```
char string[8] = "abcdefg";
*string = '\0';
printf ("%s", string);
```

   A. bcdefg   **B.** No output   **C.** Runtime Error   **D.** Compilation Error

---

8. To assign the contents of one integer array to another, you must use …

   A. the assignment operator (=) with the array names

   B. the equality operator (==) with the array names

   C. a loop to assign the elements of one array to the other array

   D. function strcpy from the standard library

---

9. Given int numbers[5]={0, 1, 2, 3, 4}; what will be the content of numbers after running the statements: int* ptr=numbers; ++*ptr; *++ptr; ++*ptr; *ptr++;

   A. 0 2 2 3 4   **B.** 1 1 2 2 4   **C.** 1 2 2 3 4   **D.** 1 2 3 4 5

---

10. Which while-loop is equivalent to (causes same side effects as) the following for-loop?
```
int i,j; for (i=0,j=5; i <5, j >0; i++,j--) printf("%d", i);
```

   A. int i=0,j=5; while (j>0){ printf("%d", i++); j--;}

   B. int i=0,j=5; while (i<5 && j>0){ printf("%d", i++); j--;}

   C. int i=0,j=5; while (i<j && j>0){ printf("%d", i); i++,j--;}

   D. int i=0,j=5; while (i<5 && j>0){ printf("%d", i++,j--);}

Question 2: Complete the code fragments so they generate exactly the output shown.

   A. printf(……………………………………………………………………, 627.14);

				6	.	2	7	1	e	+	0	2

   B. printf("……………………………………………………………, 0.888);

			+	0	.	8	9

   C. int i; for(i=………;i…………;i……………) printf("%d ",………………);

1		-	3		9		-	2	7	

   D. int i=1234; while(……………………………) printf("%d ",…………………);

1	2	3		1	2		1		0	

   E. int i=…………; while(………………………………) printf("%d ",………………);

-	5	0		0		5	0		1	0	0	

Question 3: Given the following declarations, answer the following questions:

```
float grades[6] = {6.5, 1.5, 3, 7.5, 10, 9}, f = 99;
float *pf1=grades, *pf2=&grades[5], **pf3=&pf1;
```

A. Complete the memory state below:

	grades						pf1	pf2	pf3
Address	0x2600	....				....	0x2700	0x2704	0x2708
Content	1.5					....	....	....	....

B. What is the value of each of the following expressions:

&grades[3]   ……………………..   &(*(pf1+3))   ……………………..

pf2 - pf1   ……………………..   *(*pf3+3)+3)   ……………………....

Question 4: Complete the following function `getDistinct` that takes an integer array A of size `n`, and returns an array that contains all the distinct elements in A. It also returns the size of the resulting array in an output parameter `size`.

```
int* getDistinct(int A[], int n, …………………………………… size){
  int i, j;
  int* result = …………………………………………………………………………………………………………………;
  /* initialize the size of the resulting array */

  …………………………………………………………………………………;
  if (result){
    /* copy the first element to the resulting array */

  …………………………………………………………………………………;
    for (i = 1; i < n; i++){
      for (j = 0; j < i; j++)
        /* Stop the inner loop if a redundant element is found */

        …………………………………………………………………………………;

      /* Add a new element to the resulting array if no redundancy is

        reported, and increment size */

      …………………………………………………………………………………;
    }
    /* Adjust the size of the resulting array to be exactly equal to
        the number of distinct elements */

    result = …………………………………………………………………………………………………………………;
  }
  return result;
}
```