



T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü



Dersin Kodu: BIMU1052	Dersin Adı: Introduction to Algorithms
Öğretim Üyesi: Dr. Öğr. Ü. Şafak DURUKAN ODABAŞI Dr. Öğr. Ü. Özgür Can TURNA	Sınav Türü: Final
Sınav Tarihi ve Süresi: 03. 01.2019 / 75 dk	Öğrenci No:
Öğrenci Ad-Soyad:	Öğrenci İmzası:

MULTIPLE CHOICES (20p)

- Which statement is equivalent to the following statement?
pt->x_center = 10.0;
a. *pt.x_center = 10.0;
b. (*pt).x_center = 10.0;
c. (*pt.)x_center = 10.0;
d. (pt->)x_center = 10.0;
- Which of the following is not a legal variable name?
a. _something
b. aVariable
c. float2string
d. 2manyLetters
- Suppose we want to store the value 1.5 into a double variable d. Which of the following **would not work**?
a. d = 1.5;
b. d = 3/2;
c. d = 1 + 0.5;
d. All of them work.
- Consider the following code fragment carefully.
How many times will the **cout** statement execute:

```
for (i = 0; i < 5; i++);  
    cout << i;
```


a. 5 times
b. 4 times
c. 6 times
d. 1 time
- A function prototype does not have to _____.
a. include argument names
b. end with a semi-colon
c. agree with the function header
d. match with all calls of the function

6. Which of the followings are TRUE about the null terminator is?

- used to advance to a new line when executing a cout statement
 - represented as '\0'
 - used to end a valid string
 - all of the statements are wrong.
- a. I-II b. I,II,III c. II,III d. IV

7. Which of the following fragments of C++ code will print the contents of an array called **ar** that has been defined to **hold 30 integer values** and **has been initialized with 30 integer values**?

- ```
int i;
a. cout << ar[];
b. for(i = 0; i == 30; i++)
 cout << ar[i] << " ";
c. for(i = 0; i < 30; i++);
 cout << ar[i] << " ";
d. for(i = 0; i < 30; i++)
 cout << ar[i] << " ";
```

8. Which of the following is the correct definition for a variable called **dblRef** that can hold a reference to the double variable **dblNum**?

- double dblRef = dblNum;
- double \*dblRef = &dblNum;
- double &dblRef = dblNum;
- double &dblRef;

9. (30p) Write a complete C++ code to find Armstrong numbers within an interval given by the user.  
A positive integer is called an Armstrong number if the sum of cubes of individual digit is equal to that number itself. For example:  
 $153 = 1 * 1 * 1 + 5 * 5 * 5 + 3 * 3 * 3$  // 153 is an Armstrong number.  
 $12$  is not equal to  $1 * 1 * 1 + 2 * 2 * 2$  // 12 is not an Armstrong number.

```
#include <iostream>
using namespace std;

int main()
{
 int origNum, num, rem, sum = 0;
 cout << "Enter a positive integer: ";
 cin >> origNum;

 num = origNum;

 while(num != 0)
 {
 rem = num % 10;
 sum += rem * rem * rem;
 num /= 10;
 }

 if(sum == origNum)
 cout << origNum << " is an Armstrong number.";
 else
 cout << origNum << " is not an Armstrong number.";

 return 0;
}
```

10. (10pt) Complete the following recursive C++ function to compute the Fibonacci numbers (0 1 1 2 3 5...):

```
int fibonacci (int n) {
 if((n==1) || (n==0))
 {
 return(n);
 }
 else
 {
 return(fibonacci(n-1)+fibonacci(n-2));
 }
}
```

11. (20p) Write the output of following code segments.

|                                                                                                                           |       |
|---------------------------------------------------------------------------------------------------------------------------|-------|
| a. double *pt;<br>double a[3]={1.2, 2.3, 3.4};<br>pt=&a[1];<br>pt+=1;<br>cout<<*pt<<endl;                                 | 3.4   |
| b. int x = 1, y = 0;<br>if (x > 0 && y < 0) {<br>x = y = 23;<br>}<br>cout << x << " " << y << endl;                       | 1 0   |
| c. int x = 1, y = 0;<br>if (x > 0    y < 0) {<br>x = y = 23;<br>}<br>cout << x << " " << y << endl;                       | 23 23 |
| d. x = 10; y = 40;<br>if (x >= 10)<br>if (y < 40) {<br>y++;<br>}<br>else {<br>y--;<br>}<br>cout << x << " " << y << endl; | 10 39 |

12. (20p) Write a C++ function called **findCommon()** which takes 2 strings as parameter to find common letters of these strings. Your function must print common letter(s). **You can use any string library function if you like.**

```
void findCommon(char *st1, char *st2)
{
 char arr1[256] = { 0 }; //All is set to zero
 char arr2[256] = { 0 }; //All is set to zero
 int i;
 for (i = 0; i < strlen(st1); i++) //List character frequency in first string
 arr1[st1[i]] ++;
 for (i = 0; i < strlen(st2); i++) //List character frequency in second string
 arr2[st2[i]] ++;
 for (i = 0; i < 256; i++)
 {
 if (arr1[i] > 0 && arr2[i] > 0) // if both string has thesame char print it.
 {
 cout << ((char)i) << " ";
 }
 }
}
```