CS 201 (Introduction to Computing)

Fall 2017-2018

 Sabancı University - Sample Questions and Solutions for Midterm 1

* Those questions do not imply any favorite subject or question type for the questions in the actual exam
* Please also review recitation questions, samples covered in the class, homeworks and the questions in the book as well
* The amount of questions here is of course much more than the actual exam.

##### QUESTIONS

**1)** Write a function that takes an integer type of parameter and returns its reciprocal. If the parameter is zero, then the function should return 0.

In the main program, first input two integer values. Then find the sum of the reciprocals of all of the numbers between these two inputs. In the main program, you have to use the reciprocal function.

**#include <iostream>**

**using namespace std;**

**double reciprocal(int number)**

**{**

 **if (number == 0)**

 **return 0;**

 **else**

 **return (1.0/number);**

**}**

**int main()**

**{**

 **int start,end;**

 **cout << "Enter the boundaries( start, end): ";**

 **cin >> start >> end;**

 **double sum = 0;**

 **if (start > end)**

 **cout << "Start must be smaller than end!" << endl;**

 **else**

 **{**

 **while (start <= end)**

 **{**

 **sum += reciprocal(start);**

 **start++;**

 **}**

 **cout << "Sum of reciprocals: " << sum << endl;**

 **}**

 **return 0;**

**}2)** Write a function that takes three string values as parameters and displays them in lexicographical order.

**void order(string s1, string s2, string s3)**

**{**

 **if ( s1 <= s2 && s2 <= s3)**

 **{**

 **cout << s1 <<" " << s2 <<" "<< s3 << endl;**

 **}**

 **else if ( s2 <= s1 && s1 <= s3)**

 **{**

 **cout << s2 <<" " << s1 <<" " << s3 << endl;**

 **}**

 **else if ( s3 <= s1 && s1 <= s2)**

 **{**

 **cout << s3 <<" " << s1 <<" " << s2 << endl;**

 **}**

 **else if ( s3 <= s2 && s2 <= s1)**

 **{**

 **cout << s3 <<" " << s2 << " " << s1 << endl;**

 **}**

 **else if ( s2 <= s3 && s3 <= s1)**

 **{**

 **cout << s2 << " " << s3 << " " << s1 << endl;**

 **}**

 **else if ( s1 <= s3 && s3 <= s2)**

 **{**

 **cout << s1 <<" " << s3 << " " << s2 << endl;**

 **}**

**}**

**3)**

**a)** Rewrite the following expression using C++ syntax.



( (c – (a/(b-c) ) ) / sin( b/(c/a) ) ) \* (a\*a\*a)

**b)** What are the results of the following expressions?

5+3/5+72\*!(2+5)==12%8\*-10

false or 0

(20\*1.5/(2\*6-8)-7)\*11

5.5

**c)** The range of short unsigned integer is [0 .. 65535]. Why is the upper limit is 65535?

Since it is stored in 16 bit and there are no negative numbers, the upper limit is 216-1.

**d)** Write the statement to display the sum of the digits of a two-digit integer number (call it num).

cout << (num/10 ) + (num%10) << endl;

**e)** We do not include anything to use int, but we need to include string header file to use string. Why is that?

Because int is a built-in data type. But string is defined in a library so we have to add its header file to use it.

**f)** Suppose you are writing a program to convert days into seconds. Assuming you are doing integer arithmetic using unsigned long integers for this purpose, what should be the maximum value of days to avoid any overflow problems?

(pow(2,32)-1)/(24x60x60) or UINT\_MAX / (24x60x60)

**g)** Rewrite using for statement instead of while

**while (year <= 4002)**

**{**

 **sum += year;**

 **cout << year << endl;**

 **year += sum;**

**}**

**for ( ;year <= 4002; year += sum)**

**{**

**sum += year;**

**cout << year << endl;**

**}**

**4)** Trace and display the output of the program using the data given below?

#include <iostream>

using namespace std;

int main()

{

int cnt, x, y, dif;

cnt = 0;

cin >> x;

do

 {

cin >> y;

dif = y - x;

if (dif > 1) while (x+1 < y) {

++x;

 cout << ”Number= ” << x << endl;

}

 x = y;

 ++cnt;

 } while (cnt != 3);

return 0;

}

Data input

12

15

16

20

##### Answer:

##### Trace

x y dif cnt

~~12~~ ~~15~~ ~~3~~ ~~0~~

~~13~~ ~~16~~ ~~1~~ ~~1~~

~~14~~ 20 4 ~~2~~

~~15~~ 3

~~16~~

~~17~~

~~18~~

~~19~~

20

##### Output will be

**Number =13**

**Number =14**

**Number =17**

**Number =18**

**Number =195)** What is the output of the following program?

**#include <iostream>**

**using namespace std;**

**int A(int i);**

**float B(char c);**

**void C(float x);**

**int main()**

**{**

 **int p = 0;**

 **float q = 0.0;**

 **char r = 'D';**

 **p = A(p);**

 **q = B(r);**

 **C(B(r));**

 **p = A(A(p));**

 **return 0;**

**}**

**int A(int i)**

**{**

 **cout << "A" << endl;**

 **return i+1;**

**}**

**float B(char c)**

**{**

 **cout << "B" << endl;**

 **return 7;**

**}**

**void C(float x)**

**{**

 **cout << "C" << endl;**

**}**

**Output:**

**A**

**B**

**B**

**C**

**A**

**A**

**6)** There are up to five (5) **syntax** errors of the following program. However, there might be less than five. Find those syntax errors and explain the nature of the error. Give the line numbers of the errors. If your syntax error is not actually a syntax error, that will reduce your grade!

**1 #include <iostream>**

**2 using namespace std;**

**3**

**4 void desidero (int hulahop)**

**5 {**

**6 if (hulahop)**

**7 {**

**8 cout << 10;**

**9 }**

**10 }**

**11**

**12 int main()**

**13 {**

**14 cout << "Program is now started";**

**15 int hokuspokus = 4.6, five;**

**16 double = hokuspokus/2.0;**

**17**

**18 cin >> hulahop >> five;**

**19 cin >> hokuspokus;**

**20 if (hokuspokus+72 != five) { cout << "five is "**

**21 << five;**

**22 }**

**23 return 0;**

**24 }**

**25**

**26 void function1**

**27 {**

**28 cout << 3 + 5;**

**29 }**

**Syntax Errors:**

1. Line:\_\_16\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_no variable name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Line:\_\_18\_\_ \_\_\_\_\_\_\_\_\_\_\_hulahop is undeclared\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Line:\_\_26\_\_\_ \_\_\_\_\_\_\_illegal function definition\_in function1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Line:\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Line:\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7)** XCELL telephone company is calculating the cost of a telephone conversation by multiplying the duration (in minutes) with the unit cost. The unit cost depends on the area code of the called telephone number. If the area code is between 200-299, then the unit cost per minute is 100 KTL, else if the area code is between 300 – 499, then the unit cost per minute is 150 KTL, else if the area code is between 500 – 599, then the unit cost per minute is 300 KTL.

In a full program:

Write a function that takes area code and duration as parameters and display the total cost of the conversation. Area code is assumed to be between 200 and 599 and the duration is assumed to be non-negative in this function (you will make input checks in the main).

In main, input the area code and the duration. First check validity of the input (area code must be between [200 .. 599], duration must be non-negative). If the inputs are valid, use the function to calculate the cost of conversation and display it on the screen, otherwise display an error message showing which input is invalid.

**#include <iostream>**

**using namespace std;**

**void cost(int areaCode, int duration)**

**{**

 **int cost;**

 **if (areaCode < 300)**

 **{**

 **cost = 100 \* duration;**

 **}**

 **else if (areaCode < 500)**

 **{**

 **cost = 150 \* duration;**

 **}**

 **else**

 **{**

 **cost = 300 \* duration;**

 **}**

 **cout << "The cost is : " << cost << endl;**

**}**

**void main()**

**{**

 **int areaCode, duration;**

 **cout << "Enter the areaCode and duration : ";**

 **cin >> areaCode >> duration;**

 **if ( areaCode < 200 || areaCode > 599 || duration < 0)**

 **{**

**cout << "Error: You have entered incorrect input!"<<endl;**

 **}**

 **else**

 **{**

**cost(areaCode, duration);**

 **}**

**}**

**8)** Write a program that inputs a numeric month value (between 1 and 12) and a year value. The program should calculate and display the month (again in numeric format) and year after 100 months.

As a variation, instead of 100 use another input. In this way, any amount of months could be added.

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

 **int month,year,amount;**

 **cout << "Enter month, year, and amount : ";**

 **cin >> month >> year >> amount;**

 **int totalMonth = month + amount;**

 **if ( (totalMonth%12) == 0)**

 **{**

 **year = (year + totalMonth / 12) - 1;**

 **month = 12;**

 **}**

 **else**

 **{**

 **year = year + totalMonth / 12;**

 **month = totalMonth % 12;**

 **}**

**cout << "Month : " << month << " " << "Year : " << year << endl;**

 **return 0;**

**}**

**9)** Write a program that reads a sequence of integers from keyboard and checks whether that sequence is ordered or not; Sentinel value is –1, i.e., the input entry should finish when –1 is entered and this value should not be considered in the sequence. The program should display a message about the order of the input sequence too.

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

 **int numPre, numCur;**

 **bool ordered = true;**

 **cout << "Enter the sequence of integers: ";**

 **cin >> numPre;**

 **if (numPre == -1)**

 **{**

 **cout << "List not started" << endl;**

 **}**

 **else**

 **{**

 **cin >> numCur;**

 **while (numCur != -1 && ordered)**

 **{**

 **if (numCur < numPre)**

 **{**

 **ordered = false;**

 **}**

 **else**

 **{**

 **numPre = numCur;**

 **cin >> numCur;**

 **}**

 **}**

 **if (ordered)**

 **{**

 **cout << "Ordered List" << endl;**

 **}**

 **else**

 **{**

 **cout << "Not an ordered List" << endl;**

 **}**

 **}**

 **return 0;**

**}**

**10)** Write a function that takes a string (let’s call it mystr) and an integer (let’s call it myint) as parameters. The function should move last myint characters of mystr to the beginning and return the resulting string.

For example, if mystr is “Bu sinav ne kadar da kolay” and myint is 11, then the returned string will be “ar da kolayBu sinav ne kad”

If myint is larger than the length of mystr, then the function should return mystr unchanged.

Remark: You can write this function with or without using loops. Do it in both ways just to practice.

**string f1(string mystr, int myint)**

**{**

 **string result = mystr;**

 **int strLen= mystr.length();**

 **int pos;**

 **if (myint < strLen)**

 **{**

 **pos = strLen - myint;**

 **result = mystr.substr(pos, myint) + mystr.substr(0, pos);**

 **}**

 **return result;**

**}**

**string f2(string mystr, int myint)**

**{**

 **string result = mystr;**

 **int ctr = 0;**

 **int strLen = mystr.length();**

 **if (myint < strLen)**

 **{**

 **result = "";**

 **while (ctr < myint)**

 **{**

 **ctr++;**

 **result = mystr.substr(strLen-ctr, 1) + result;**

 **}**

 **for (ctr=0; ctr < strLen-myint; ctr++)**

 **{**

 **result = result + mystr.substr(ctr, 1);**

 **}**

 **}**

 **return result;**

**}**

**11)** Write a function that takes an integer parameter (say *N*) and displays the positive powers of 2 that are smaller than *N*. For example if *N* is 3000, then the function displays

1 2 4 8 16 32 64 128 256 512 1024 2048

**void power(int N)**

**{**

 **int ctr = 0,power = 1;**

 **while (power < N)**

 **{**

 **cout << power << " ";**

 **ctr += 1;**

 **power = pow(2, ctr);**

 **}**

**}**

**12)** **Use only unsigned integers in this question**.

In a program

Write a function that takes a volume value in cubic meters as parameter and displays its equivalent in cubic centimeters (one cubic meter = 1,000,000 cubic centimeters)

In main, input the cubic meter information and call the above function to display this volume in cubic centimeters. Before calling the function, check the input so that the result should not overflow. To perform such a check, first you should calculate the maximum value for the cubic meters input and then enforce the input in that limit.

**#include <iostream>**

**#include <climits>**

**using namespace std;**

**void convert(unsigned int m3)**

**{**

 **cout << m3 \* 1000000 << endl;**

**}**

**int main()**

**{**

 **unsigned int m3;**

 **unsigned int limit = UINT\_MAX / 1000000;**

 **cout<<"Enter cubic-meter - "<<endl<<"limit is : "<<limit<<" : ";**

 **do**

 **{**

 **cin >> m3;**

 **if (m3 <= limit)**

 **convert(m3);**

 **else**

 **cout << "OUT OF LIMIT. TRY AGAIN" << endl;**

 **} while (m3 > limit);**

 **return 0;**

**}**

**13)** Write a program that reads a string from keyboard and displays the characters between the last two commas of that string. Do not display the commas. If the input string has only one comma, then display all the characters after it. If the string does not have any commas, then display the whole string.

You can write this program with or without loops. Do it in both ways just to practice.

**#include <iostream>**

**#include <string>**

**using namespace std;**

**//if solution**

**int main()**

**{**

 **string original, untillastcomma;**

 **unsigned int lastcommapos, lastcommapos2;**

 **cout << "Please enter your string: ";**

 **cin >> original; //input the string**

 **lastcommapos = original.rfind(",");**

 **//find the position of last comma**

 **if (lastcommapos != string::npos) //if comma exists**

 **{**

 **untillastcomma = original.substr(0, lastcommapos);**

 **//extract all the characters until last comma**

 **}**

 **else //if no comma exists**

 **{**

 **untillastcomma = original; //use the original string**

 **}**

 **lastcommapos2 = untillastcomma.rfind(",");**

**//find the last comma in the extracted string**

 **//that corresponds to the second comma from the back**

 **//in the original string**

 **if (lastcommapos2 != string::npos)**

**//if second comma exists**

 **{**

**cout << untillastcomma.substr(lastcommapos2+1, untillastcomma.length()) << endl;**

 **}**

 **else if (lastcommapos != string::npos)**

**//if there is only one comma in the original string**

**{**

**cout<<original.substr(lastcommapos+1,original.length())<<endl;**

 **}**

 **else //if no comma exists**

 **{**

 **cout << original << endl;**

 **}**

 **return 0;**

**}**

**//Another (alternative) if solution**

**int main()**

**{**

 **string original, resultString;**

 **unsigned int lastcommapos,untillastcomma;**

 **cout << "Please enter your string: ";**

 **cin >> original;**

 **lastcommapos = original.rfind(",");**

 **if (lastcommapos != string::npos)**

 **{**

 **untillastcomma = original.rfind(",", lastcommapos-1);**

 **if(untillastcomma == string::npos)**

 **{**

 **resultString = original.substr**

 **(lastcommapos+1,original.length());**

 **}**

 **else**

 **{**

**resultString = original.substr(untillastcomma+1,**

 **lastcommapos-untillastcomma-1);**

 **}**

 **}**

 **else**

 **{**

 **resultString = original;**

 **}**

 **cout <<resultString <<endl;**

 **return 0;**

**}**

**//alternative solution with loop**

**int main()**

**{**

 **string original, resultString;**

 **int lastcommapos = -1,prevCommaPos = -1;**

 **cout << "Please enter your string: ";**

 **cin >> original;**

 **int index=0;**

 **while (index < original.length())**

 **{**

 **int temp;**

 **temp = original.find(",",index);**

 **if (temp != string::npos)**

 **{**

 **prevCommaPos = lastcommapos;**

 **lastcommapos = temp;**

 **index = temp + 1;**

 **}**

 **else**

 **index = original.length();**

 **}**

 if (prevCommaPos == -1)

 {

 if (lastcommapos == -1)

 {

 cout <<original<<endl;

 }

 else

 {

 Cout << original.substr

 (lastcommapos+1,original.length()) << endl;

 }

 }

 else

 {

 cout << original.substr(prevCommaPos+1,

 lastcommapos - prevCommaPos-1)

<< endl;

 }

 return 0;

}

**14)** Write a function that takes three string parameters (let’s call them s1, s2 and s3) and replaces all of the occurrences of s2 in s1 with s3. The function should return the resulting string. For example, if s1 is “hatasiz sinav olmaz, hatamla sev beni”, s2 is “hata” and s3 is “kopya”, then the function returns “kopyasiz sinav olmaz, kopyamla sev beni”.

string changeWord(string s1, string s2, string s3)

{

 unsigned int index=0, length, s2\_len, s3\_len;

 length = s1.length();

 s2\_len = s2.length();

 s3\_len = s3.length();

 while (index < length)

 {

 index = s1.find(s2, index);

 // find the start position of the s2 in s1

 if (index != string::npos)

 {

 // change string s2 with s3 in s1

 s1 = s1.substr(0,index) + s3 +

s1.substr(index + s2\_len, length);

 index += s3\_len;

 length = s1.length();

 }

 }

 return s1;

}**18)** Write a function, called findPValue, to calculate and return the p(x,n,p) given below. x is a parameter of type double, n and p are parameters of type int.



double findPValue(double x, int n, int p)

{

 double result=1.0;

 for (int i=3; i <= n; i++)

 {

 double sumResult=0;

 for(int j=i; j <= p; j++)

 {

 sumResult += fabs(x-pow(j,2));

 }

 result \*= x \* pow(sumResult, i);

 }

 return result;

}

**19)** Write a program that reads 200 integer numbers and counts the total number of positive numbers among them.

#include <iostream>

using namespace std;

int main()

{

 int num, posCtr=0;

 cout << "Enter 200 inter numbers" << endl;

 for(int i=0; i < 200; i++)

 {

 cin >> num;

 if (num > 0)

 {

 posCtr++;

 }

 }

 cout << "Total Number of Positives: " << posCtr << endl;

 return 0;

}

**20)** We have seen a palindrome check function in class. Rewrite the palindrome check function, but this time without reversing the parameter string.

Hint: in a loop, check the letter pairs that must be the same to be a palindrome.

bool checkPalindrome(string input)

{

 int length = input.length();

 int index = 0;

 while (index < length/2)

 {

 if ( input.at(index)!= input.at(length-1-index) )

 {

 return false;

 }

 index++;

 }

 return true;

}