CMPS 252::HW1

Due: Monday, Sep 20, 8:00 AM

Must be done individually.

1. Write a C++ program that produces the following output (hints: use std::hex, std::dec, std::oct, std::bitset, research using them on your own): (10 points)

dec hex oct binary : ASCII

65 0x41 O101 b01000001: A

66 0x42 O102 b01000010: B

67 0x43 O103 b01000011: C

68 0x44 O104 b01000100: D

69 0x45 O105 b01000101: E

70 0x46 O106 b01000110: F

71 0x47 O107 b01000111: G

72 0x48 O110 b01001000: H

73 0x49 O111 b01001001: I

74 0x4a O112 b01001010: J

75 0x4b O113 b01001011: K

76 0x4c O114 b01001100: L

77 0x4d O115 b01001101: M

78 0x4e O116 b01001110: N

79 0x4f O117 b01001111: O

80 0x50 O120 b01010000: P

81 0x51 O121 b01010001: Q

82 0x52 O122 b01010010: R

83 0x53 O123 b01010011: S

84 0x54 O124 b01010100: T

85 0x55 O125 b01010101: U

86 0x56 O126 b01010110: V

87 0x57 O127 b01010111: W

88 0x58 O130 b01011000: X

89 0x59 O131 b01011001: Y

90 0x5a O132 b01011010: Z

#include <iostream>

#include <bitset>

int main()

{

std:: cout<< "dec " << " hex " <<"oct " << " binary " << "Ascii" <<std::endl;

for ( int i = 65; i <=90 ; i++) {

const int a =i;

std::cout << std::dec << i << ' '

<< std::hex << "0\*" <<i << ' '

<< std::oct << i << ' '

<< std::bitset<8>{static\_cast<unsigned long long>(a)} << ' '

<< (char) i << std::endl;

}

}

1. What is the output of this code? (3 points)

int x = 78;

int y = 12;

int\* p = &x;

int\* q = &y;

p = q;

\*p = 90;

cout << x << " " << y << endl;

cout << \*p << " " << \*q << endl;

the out put will be of order 78 90 and 90 90. The variable x does not change in

content and though initially the value of the pointer pointed to x it was later reasigned to

point to y hence the result.

Note: though the standard header was included, specificity on the class was still needed

1. Given this code segment: (3 points)

int i = 5;

int\* r = &i;

int\*\* s = &r;

std::cout << "the value of s is " << \*\*s << std::endl; // prints 5

int\*\*\* t = &s;

std::cout << "the value of t is " << \*\*\*t << std::endl; // prints 5

1. Add a variable r and store the address of i in r
2. add a variable s and store the address of r in s
3. use s to print the value of i (5)
4. add a variable t and store the address of s in t
5. use t to print the value of i (5)
6. Fill in the blank (5 points)

int nums[]{ 10,20,30,40,50,60,70,80,90 };

cout << nums << endl; //record the value you get: 010FFB28

//explain why you got such a value: nums points to a memory address of type in that houses

// the elements of the array( it doesn't point to any value)

cout << \*nums << endl;//record the value you get: 10

//explain why you got such a value: a pointer to an array(without using size of the array

//e.g int (\*ptr)[5]) points to the first element of the array

cout << \*nums+1 << endl;//record the value you get: 11

//explain why you got such a value: since a pointer to an array point to the first element, // this statement gets the first value(10) and adds 1 to it

cout << \*(nums + 1) << endl;//record the value you get: 20

//explain why you got such a value: the pointer points to the next element after the first

// like indexing

int std::size(nums)-1;//assign a new value to i so that the following statement prints the last element of the array

cout << \*(nums + i);

1. The following code is supposed to reverse the elements of the array, but it has three bugs. Fix it so it reverses the array properly. (3 points)

int nums[]{ 10,20,30,40,50,60,70,80,90 };

for (size\_t i = 0; i < size(nums); i++)

{

int j = nums[i];

nums[i] = nums[size(nums) – i-1];

nums[size(nums) – i-1] = j;

}

for (int i : nums) cout << i << ' ';

1. Given the following code, answer the questions that are embedded in the comments (2 points)

std::string strings[] { "CMPS 252", "CMPS 272" };

cout << strings[1][6] << endl; //what will this print?: 7

//explain why you got such an output: strings[1] points to the second element in the array( “CMPS 272”) and strings[1][6] points to the 6th char which is 7

1. Given the following code, answer the questions that are embedded in the comments (2 points)

std::string strings[] { "CMPS 252", "CMPS 272" };

cout << (\*strings)[6];//what will this print? // : 5

//explain why you got such an output : a pointer to an array without indicating(the size of the array in the pointer) points to the first element of the array. Hence,

(\*strings)[6] points to the 6th char in the first element of the array.

1. What is the data type of the variable c? Explain your answer. (2 points)

std:string courses = "CMPS 252";

auto c = &courses[0];

// c is of type char\*. This is because the type on the RHS(&courses[0]) is char and the memory type is a char since we need a pointer to hold the address of a char we get

Char \* c

1. What will this code print? Explain your answer. (2 points)

int i = 5;

int j = 10;

int\* r = &i;

int\* s = &j;

\*r = \*s;

cout << i;

// the output is 10. r points to the address of I and s to that of j. dereferencing r now points to the actual position of I to which the j is given(\*s which points to it)

1. What will this code print? Explain your answer. (2 points)

int i = 5;

int j = 10;

int\* r = &i;

int\* s = &j;

r = s;

cout << \*r;

// when we use r=s we make r point to the value of j whose address is stored in s

1. Given a text file constitution.txt
2. Write a program that prints the words that are at least 5 characters long and occur at least 10 times. You must use a std::map to store the words and their respective occurrence (20 points)

**Sample output**

Article: 101

Budget: 10

Chamber: 87

Chapter: 11

Constitution: 16

Constitutional: 12

Council: 42

Deputies: 26

…

1. Write a program that prints the words that occur the same number of times in the file (20 points)

**Sample output**

10: Budget|Supreme|amendment|based|bills|chamber|determined|during|following|other|

11: Chapter|Power|Prime|legislative|under|

12: Constitutional|Minister|freedom|two-thirds|

13: before|extraordinary|government|members|

14: after|proposal|removal|

1. Given a text file lebanon-covid.csv write a program that reads the file and lets the user interact with the data using the following commands: (30 points)
2. Print the average number of daily new cases

Average number of daily new cases: 1075

1. Print the average number of new cases per month per year

Average number of new cases per month per year:

2/2020: 0.444444

3/2020: 15.0323

4/2020: 8.5

5/2020: 15.9677

6/2020: 18.6

7/2020: 89.5806

8/2020: 411.387

9/2020: 744.2

10/2020: 1341.74

11/2020: 1557.2

12/2020: 1727.71

1/2021: 3856.42

2/2021: 2642.79

3/2021: 3011.29

4/2021: 1939.27

5/2021: 445.484

6/2021: 149.267

7/2021: 532.71

8/2021: 1318.9

9/2021: 901.231

1. Print the average number of daily deaths

Average number of daily deaths: 14.83

1. Print average number of deaths per month per year

Average number of deaths per month per year:

3/2020: 1.03226

4/2020: 0.4

5/2020: 0.0967742

6/2020: 0.233333

7/2020: 0.870968

8/2020: 3.41935

9/2020: 6.66667

10/2020: 8.70968

11/2020: 12.7

12/2020: 14.5161

1/2021: 52.0645

2/2021: 57.5

3/2021: 49.7419

4/2021: 34.8

5/2021: 14.5484

6/2021: 4.06667

7/2021: 1.77419

8/2021: 4.74194

9/2021: 11.4615