COSC 1550 Programming 1

Practice Exam

Please record the time you used for these questions.

The time limit for the final exam is about 80~100 minutes.

So please try to complete this practice exam in 90 minutes.

Please complete this practice exam before next Thursday.

Part I

Predict the output of the following code segments. (20%)

(1)

int x = 10,

y = 15;

double a = 2.0,

b;

b = x / y \* a;

cout << b << endl;

(2)

int x = 10,

y = 15;

double a = 2.0,

b;

b = a \* x / y;

cout << b << endl;

(3)

int a = 10,

b = 20;

if (a == 10)

{

cout << 'A';

}

else if (b == 20)

{

cout << 'B';

}

else

{

cout << '?';

}

(4)

int a = 5 / 2;

switch (a)

{

case 0:

cout << 'a';

break;

case 1:

cout << 'b';

break;

default:

cout << 'c';

}

(5)

const int WIDTH = 3,

HEIGHT = 4;

int matrix[HEIGHT][WIDTH] = { { 1, 2, 3},

{ 4, 5, 6},

{ 7, 8, 9},

{10,11,12}};

cout << matrix[2][1] << endl;

cout << matrix[3][2] << endl;

(6)

for (int x = 0; x < 3; x++)

{

for (int y = 7; y > 5; y--)

{

cout << x << y << endl;

}

}

(7)

int a = 1;

while (a < 10)

{

if (a % 3 == 0)

{

a \*= 3;

}

else

{

a++;

}

cout << a << endl;

}

(8)

int sum = 0,

numbers[8] = { 3, 6, 8, 2, 5, 8, 7, 1 };

for (int x = 2; x < 5; x++)

{

sum += numbers[x];

}

cout << sum;

(9)

int x = -1,

arr[7] = { 9, 7, 6, 3, 8, 2, 5 };

for (int i = 0; i < 7; i++)

{

if (arr[i] % 2 == 0)

{

x = i;

}

}

cout << x;

(10)

int x = 1,

y = 2;

do

{

int x = 0;

x = x + y;

y--;

} while (y > 0);

cout << x;

Part II

Searching and sorting

(1)

Given these numbers in an array: 112, 135, 256, 257, 355, 364, 397, 401

If we use binary search to search for the target number 255,

show the steps of the comparisons. (5%)

(E.g. the first step is we compare 257 with 255)

(2)

Given these numbers in an array: 2, 4, 1, 3, 5

If we use bubble sort to sort this array in ascending order,

show the steps of how the array changes. (5%)

(3)

Complete the function of linear search (5%)

#inlcude <iostream>

using namespace std;

int search( ); // fill the blank

int main()

{

const int SIZE = 9;

int targetindex;

char target = 'G',

letterset[SIZE] = {'B', 'M', 'P', 'J', 'P', 'G', 'T', 'I', 'F'};

targetindex = search( ); // fill the blank

if (targetindex == -1)

cout << "The target is not found." << endl;

else

cout << "The target is found at location "   
 << targetindex + 1 << endl;

return 0;

}

int search( ) // complete the function

{

Part III

Write functions. (20%)

Write a function (function definition) for each of the questions below.

For example, write a function to compute and return the sum of 2 floating point numbers.

float computeSum(float x, float y)

{

return x + y;

}

(1)

Write a function to compute the area of a rectangle. The function takes 2 double parameters:

width and height. Then the function computes the area (width x height) and return it.

(2)

Write a function. The function asks the user to type a floating point number, and then "return"

its square and cube.

(3)

Write a function. The function takes an integer array as its parameter, and it returns the number of odd numbers in the array.

For example, if the array is {5, 6, 7, 8, 9}, the function returns 3, because there are 3 odd numbers: 5, 7, and 9. (Hint: How can the function know the size of the array?)

(4)

Write a function. The function takes 2 float arrays as it parameters. (The two arrays may have different sizes.) Then the function shall return the total sum of all elements of the arrays.

For example, if the arrays are {1, 2, 3} and {4, 5}, the function returns 15.

Part IV

Tracing functions

Predict the output of the following programs. (20%)

(1)

#include <iostream>

using namespace std;

int dosomething1(int a, int b, int c);

int dosomething2(int a, int b);

int main()

{

int a = 5,

b = 4,

c = 3,

d;

d = dosomething1(3, 2, 1);

cout << d << endl;

d = dosomething1(d, a + 3, b);

cout << d << endl;

d = dosomething1(a + 1, b, c - 1);

cout << d << endl;

return 0;

}

int dosomething1(int a, int b, int c)

{

return a + dosomething2(b, c);

}

int dosomething2(int a, int b)

{

return a / b;

}

(2)

#include <iostream>

using namespace std;

void change\_reference(int &, int);

int main()

{

int x = 5,

y = 7;

change\_reference(x, y);

cout << x << y << endl;

change\_reference(y, x); // read carefully!

cout << x << y << endl;

return 0;

}

void change\_reference(int & x, int y)

{

x++;

y--;

cout << x << y << endl;

}

(3)

#include <iostream>

using namespace std;

void change\_array(int arr[], int size);

int main()

{

const int LENGTH = 5;

int arr[LENGTH] = {5, 4, 3, 2, 1};

change\_array(arr, 3);

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

{

cout << arr[i] << endl;

}

return 0;

}

void change\_array(int arr[], int size)

{

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

{

arr[i] \*= 2;

}

}

(4)

#include <iostream>

using namespace std;

void copy\_array(int arr1[], int arr2[], int size);

int main()

{

int arr[5] = {5, 4, 3, 2, 1},

num[7] = {3, 4, 5, 6, 7, 8, 9};

copy\_array(arr, num, 4);

for (int i = 0; i < 7; i++)

{

cout << num[i] << endl;

}

return 0;

}

void copy\_array(int arr1[], int arr2[], int size)

{

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

{

arr2[i] = arr1[i];

}

}

Part V

Writing a program.

The program has an array price that holds the unit price of each fruit. price[0] is the unit price of apple. price[1] is the unit price of banana. price[2] is the unit price of grapefruit. price[3] is the unit price of watermelon. The price of each fruit is fixed.

The program has an array quantity that holds the quantity of each fruit. quantity[0] is the quantity of apple. quantity[1] is the quantity of banana. quantity[2] is the quantity of grapefruit. quantity[3] is quantity of watermelon. The quantity of each fruit is to be entered by the user.

Write a program to compute the total price of these 4 fruits.

Requirements:

(1) Your program should have a function get\_quantity(), which asks the user to type the quantities of the 4 fruits. (10%)

(2) Your program should have a function compute\_price(), which computes the total price of the 4 fruits. The function should take the above 2 arrays as the parameter and return the total price. (10%)

(3) The main function should call the above 2 functions and then display the total price on the command prompt. (5%)

You do not need to consider the situation when the user types a negative number or a non-numerical value.

Part of the program has been provided:

#include <iostream>

using namespace std;

get\_quantity( );

compute\_price( );

int main()

{

const int NUM\_OF\_FRUITS = 4;

const double price[NUM\_OF\_FRUITS] = {1.49, 0.35, 0.99, 4.99};

double totalprice;

int quantity[NUM\_OF\_FRUITS];