|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course Name:** | **Programming Fundamentals** | **Course Code:** | **CS** |
| **Program:** | **CS** | **Semester:** | **Fall 2018** |
| **Duration:** | **3 Hrs** | **Total Marks:** | **50** |
| **Paper Date:** | **25 Dec 2018** | **Weight** | **45** |
| **Section:** | **ALL** | **Page(s):** | **6** |
| **Exam Type:** | **Final Exam** |  |  |
| **Student : Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll No.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section:\_\_\_\_\_\_\_** | | | | |
| **Instruction/Notes:** | Solve the exam on this question paper. You may use rough sheets, but they must not be attached. | | | |

**Question 1** **10 Points**

There is a function available in <cmath> that allows a user to compute for any real number . However, this can also be computed using the Taylor approximation for . This is an infinite series given below:

Implement a function mySin that takes in two parameters: a real number, and an integer. The integer specifies the number of terms of the Taylor series that will be used to compute the result. For example, in the series above we’re showing the first 4 terms only. If the value of , then simply use the first 10 terms of the series to compute .

**float mySin(float x, int k)**

**{**

**if(k<1)**

**k=10;**

**int sum = 0, fact, odd=1;**

**for(int i=1;i<=k;i++)**

**{**

**int fact = 1;**

**for(int j=2;j<=odd;j++)**

**fact = fact \* j;**

**if(i%2==1)**

**sum = sum + pow(x,odd)/fact;**

**else**

**sum = sum - pow(x,odd)/fact;**

**fact=1;**

**odd+=2;**

**}**

**return sum;**

**}**

**Question 2** Pascal’s triangle is a triangle of numbers whose first five rows (row 0 to row 4) are shown below:

**4+3+3 Points**

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

On careful observation, you will see that each row begins and ends with a 1. Therefore, as you can see, the first row contains 1; the second row contains two 1’s. After this, every row starts and ends with a 1, and the terms in between are calculated by taking the sum of two consecutive terms in the previous row.

Your task is to write code that generates a Pascal’s triangle, flips it upside down, and then destroys it as per instructions given below.

void main()

{

int n;

cout<<”Enter Rows”;

cin>>n;

/\* a. Write code here that dynamically creates a two-D array consisting of rows to store Pascal’s triangle on heap. Each row must be **exactly** the same size as the number of elements in that row. Populate the array with appropriate values. If the value of n is 5, then this code should create the triangle given above \*/

int \*\* triangle;

if (n > 0){

triangle = new int\*[n];

for (int i = 0; i < n; i++){

triangle[i] = new int[i + 1];

triangle[i][0] = triangle[i][i] = 1;

for (int j = 1; j < i; j++)

triangle[i][j] = triangle[i-1][j - 1] + triangle[i-1][j];

}

}

/\* b. Write code here inverts/flips the triangle top down, such that the first row becomes the last row, etc. This function must be efficient that is it must not move individual items/terms to perform this inversion but only move pointers. As an example, after inversion the above triangle must become:

1 4 6 4 1

1 3 3 1

1 2 1

1 1

1

\*/

int \* temp = nullptr;

for (int i = 0; i <= n/2; i++){

temp = triangle[i];

triangle[i] = triangle[n-i-1];

triangle[n - i - 1] = temp;

}

/\* c. Finally write code here that de-allocates the triangle. Make sure there are no leaks. /\*

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

delete [] triangle[i];

delete [] triangle;

triangle = nullptr;

**Question 3 10 Points**

Write a function findStrInMat that finds a cstring (null terminated char array) in a 2D character matrix of size 8x8. The string may only appear in the matrix horizontally, however, it is possible for a string to wrap around from the end of a row to the beginning of the next row. It is also possible for the string to wrap around from the end of the last row to the beginning of the first. The function should return the number of times occurs in . In the following example the string to be found is *oxen*, and it occurs in the matrix 3 times.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **e** | **n** | j | c | y | **o** | **x** | **e** |
| **n** | t | h | f | o | x | t | u |
| m | y | j | n | e | x | o | z |
| g | h | v | d | i | r | t | n |
| u | x | **o** | **x** | **e** | **n** | b | a |
| t | i | t | o | u | t | i | r |
| t | t | g | g | h | s | o | x |
| y | u | f | g | h | s | **o** | **x** |

const int row = 8;

const int col = 8;

int findStrInMat(char mat[row][col], char str[]){

int count = 0;

char stop = false;

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

{

for (int j = 0; j < col; j++)

{

if (mat[i][j] == str[0])

{

stop = false;

int k = 1;

int ii = i;

int jj = j;

while (str[k] != '\0' && !stop)

{

if (jj == col - 1){

jj = 0;

ii++;

}

else

jj++;

if (mat[ii%(row)][jj] == str[k])

k++;

else

stop = true;

}

if (k == strlen(str))

count++;

}

}

}

return count;

}

void main(){

char mat[8][8] = { { 'e', 'n', 'j', 'c', 'y', 'o', 'x', 'e' },

{ 'n', 't', 'h', 'f', 'o', 'x', 't', 'u' },

{ 'm', 'y', 'j', 'n', 'e', 'x', 'o', 'z' },

{ 'g', 'h', 'v', 'd', 'i', 'r', 't', 'n' },

{ 'u', 'x', 'o', 'x', 'e', 'n', 'b', 'a' },

{ 't', 'i', 't', 'o', 'u', 't', 'i', 'r' },

{ 't', 't', 'g', 'g', 'h', 's', 'o', 'x' },

{ 'e', 'd', 'f', 'g', 'h', 's', 'o', 'x' }};

cout << "Count of String: " << findStrInMat(mat, "oxen") << endl;

system("pause");

}

**Question 4** What is the output of the given program? (Ignore header files etc.) **5 Points**

|  |  |
| --- | --- |
| int GuessWhat(int&a, int&b) {  int c = 0;  a = c + 1;  b = c + 2;  return a + b;  } | int GuessAgain(int&a, int&b, int c, int d) {  int i;  c = c + 5;  d = d \* 2;  for (i = 0; i< c; i++)  a = a + d;  b = c + d;  return i;  } |
| void main()  {  int a =0, b = 0, c = 0, d = 0, e = 0;  e = GuessWhat(a, b);  cout<<"The 1st Value of E is" << e;  e = GuessWhat(d, c);  cout<<"\nThe 2nd Value of E is" << e;  e = GuessAgain(a, b, c, d);  cout<<"\nThe 3rd Value of E is" << e;  e = GuessAgain(d, c, b, a);  cout<<"\nThe 4th Value of E is" << e;  \_getch();  } | **Output:**  **The 1st Value of E is3**  **The 2nd Value of E is3**  **The 3rd Value of E is7**  **The 4th Value of E is14** |

**Question 5:** Are the following pieces of code free of runtime errors? If so, write the outputs, otherwise point out the errors by encircling the lines and rewrite only corrected lines of code. **5+5 Points**

|  |  |
| --- | --- |
| a) void main(){ |  |
| int \*\* a = new int\*[3]; |  |
| int \*\* a2 = new int\*[3]; |
| for (int i = 0; i < 3; i++){ |  |
| a[i] = 0; |
| a2[i] = 0; |
| } |  |
| a[0] = new int(40); // **ALERT!** a[0] is allocating an (one) integer and it is initialized to 50. We are **not** creating an array of 40 integers. |  |
| a[1] = new int(100); //same **ALERT!** |  |
| a2[1] = new int(50); //same **ALERT!** |
| a2[2] = a[1]; |
| for (int i = 0; i < 3; i++){ | for (int i = 0; i < 3; i++){  if (a2[i] != 0){  for (int j = 0; j < 3; j++){  if (a2[i] == a[j])  a2[i] = 0;  }  }  }  for (int i = 0; i < 3; i++){  if (a[i] != 0) delete a[i];  if (a2[i] != 0) delete a2[i];  } |
| if (a[i] != 0) |
| cout<< \*a[i] <<" "; |
| if (a2[i] != 0) |
| cout << \*a2[i] << " "; |
| } |
| for (int i = 0; i < 3; i++){ |
| delete a[i]; |
| delete a2[i]; |
| } |
| delete[] a; |  |
| delete[] a2; |
| a = 0; |
| a2 = 0; |
| } |  |

|  |  |
| --- | --- |
| b) |  |
| int main(){ |  |
| char \*\* mypets = new char\*[2]; |  |
| char \* Cat = new char[50]; |
| char \* Dog = new char[50]; |
| strcpy(Cat, "Milo is a Furry Cat\n"); |  |
| strcpy(Dog, "Courage is a Brave Dog\n"); |  |
| mypets[0] = Cat; |  |
| mypets[1] = Dog; |  |
| delete[] Cat; | **Remove** |
| for (int i = 0; i < 2; i++) |  |
| cout << mypets[i] << endl; |
| delete[] mypets; | **Add** delete[] Cat; delete[] Dog; |
| mypets = nullptr; |  |
| cout << endl; |  |
| return 0; |  |
| } |  |

**Question 6 2+2+1 Points**

1. Create a dynamic 2D character array of dimensions 5x6.

**char \*\*data = new char\*[5];**

**for(int i=0;i<5;i++)**

**data[i] = new char[6];**

1. main() {

int\*new1 = 0;

int a;

cin>>a;

/\*print the value of *a* without using the variable *a* in the cout statement. You maybe not declare any new variables to do this. Write the code only in the box below.\*/

**new 1 =&a;**

**cout<<\*new1**;

\_getch();

}

1. Consider the two functions:

int find(int data[],int size, int key)// finds and returns the index of *key* in array *data*.

void replace(int data[],int size, int val, int i) /\*stores the value *val* at index *i* in array *data*.\*/

Now consider the following code:

int main(){

int somenumbers[10] = { … }; //initialized with some values, in which one, and exactly one

// value is 6.

// Write a **single** c++ statement that would replace the element 6 in the array with the value 30 using the // 2 functions given above.

//single statement here:

**replace(somenumbers, 10, 30, find(somenumbers, 10, 6));**

} /end of main