CS-1004 Object Oriented Programming Spring-2023
ASSIGNMENT-01

Section (All)

Submission Deadline: 23rd February, 12:30 pm.

Instructions:

1. Assignments are to be done individually. You must complete this assignment by yourself. You cannot work with anyone else in the class or with someone outside of the class. The code you write must be your own and you must understand each part of your code. You are encouraged to get help from the instructional staff through google classroom.
2. Do not use any String or math libraries (such as cmath, cstring, string etc) and also do not use built-in function (such as strlen, strcmp etc). **Caution**: zero marks will be awarded.
3. Do not edit **Function Prototypes**. **Caution**: zero marks will be awarded.
4. The usage of string is strictly prohibited.
5. Your code must be **generic , use dynamically created arrays only** .
6. Marks distribution and test Cases are provided for each question. Your code will be evaluated with **similar test cases**. If the required output is generated, you will be awarded full marks. Failing to generate the correct output will result in zero marks. Total Marks: 200.
7. For **PrintPattern and Snake Game** questions, the output should be properly displayed and well presented. There will be no gtests for **PrintPattern and Snake Game** questions. Static and global variables not allowed for recursive functions.
8. **Plagiarism**: Plagiarism of any kind (copying from others, copying from the internet, etc) is not allowed. If found plagiarized, you will be awarded zero marks in the assignment. Repeating such an act can lead to strict disciplinary actions and failure in the course.
9. **Please start early otherwise you will struggle with the assignment.**
10. **Test cases:** Test cases (in gtest) will be shared with you on Google Classroom. **We will be running your code against our test cases, and a test case failure or a segmentation fault/incorrect result or even syntax error will result in zero marks.**
11. **Submission Guidelines**: Dear students, we will be using **auto-grading tools (gtest)**, so failure to submit according to the below format would result in zero marks in the relevant evaluation instrument.
    a. Make your own file named submission.cpp. Please don't include the main function while submitting the file. And don't remove **test cases** (in testcases.cpp) or **function prototypes** (in submission.cpp).
    b. Your submission.cpp file must contain your name, student-id, and assignment # on the top of the file in the comments.
    c. Move you submission.cpp in one folder. The folder must contain only submission.cpp file (no binaries, no exe files etc.,). If we are unable to download your submission due to any reason you will be awarded zero mark.
    d. Run and test your program on a lab machine before submission. If there is a syntax error, zero marks will be awarded in that specific question.
    e. Rename the folder as ROLL-NUM_SECTION (e.g. 22i-0001_A) and compress the folder as a zip file. (e.g. 22i-0001_A.zip). Only zip file will be acceptable.
    f. Submit the .zip file on a form shared with you along with the assignment within the deadline.
    g. Submission other than that form (e.g. email etc.) will not be accepted.

h. The student is solely responsible to check the final zip files for issues like corrupt files, viruses in the file, mistakenly exe sent. If we cannot test the file due to any reason it will lead to zero marks in the assignment.

Note: Follow the given instruction to the letter, failing to do so will result in a zero.

Q1: String and array Manipulation

Note: All these questions are to be done using character arrays and string datatype is not allowed

Marks: 10

1. Given a paragraph as a character array, write a c++ program to remove a substring from that paragraph. Your function should return the original array after removing the input array.

   Function Prototype:
   char removeSentence(char* Para, char* input);*

   For Example:
   Consider the following paragraph,
   "I am currently studying OOP course. I hope to pass it. I might fail."
   Input: "I hope to pass it."
   Resulting paragraph will be as follows:
   "I am currently studying OOP course. I might fail."

Marks: 10

2. Given a char array, find if a subarray exists in that array. The array has to be treated as **circular**. Your function should return a Boolean true or false.

   Function Prototype:
   bool FindSubString(char Str, char* substr,int & start, int & end);*

   For Example:
   Consider the following array,
   "abcdab"
   Input: "cda"
   Output : return true , start =2 , end=4
   Input: "dababc"
   Output : return true , start =3 , end=2
   Input: "bad"
   Output : return false , start =-1 , end=-1

Marks: 50
Q2: 3d String Matching

Write a C++ program to find a string in a 3d character matrix. Your function should check whether a string matches and return an 2d array which saves matched coordinates (start till ending) in the matrix. The matching can be done on each slice of the 3D array (row, column, diagonal direction).

Note: slices can be in x, y and z direction

For example,
Consider a 3d matrix of 3x3x4 size as shown below

a	b	a	'\0'
x	o	x	'\0'
p	a	t	'\0'

p	a	k	'\0'
m	a	n	'\0'
d	a	y	'\0'

n	t	s	'\0'
f	a	t	'\0'
s	a	s	'\0'

Let the input string be "man",
Match found "True"
2 D array

X=>	1	1	1
Y=>	1	1	1
Z=>	0	1	2

Similarly, if the string is "bat", the output will be :
Match found "True"
2 D array

X=>	0	1	2
Y=>	0	0	0
Z=>	1	1	1

if the string is "ms", the output will be :
Match found "True"
2 D array

X=>	1	2
Y=>	1	2
Z=>	0	0

If no strings are found, the resulting 2D array will have -1 sizes and return false.
The **functions prototype** are as follows:

char*** ConvertToDynamic(char arr[], int x, int y, int z);
(This function takes a 1d array as argument and the 3 sizes and creates and return a 3d array)

bool MatchString3DArray(char*** mat, int xSize, int ySize, int zSize, char * input, int**& resultMat, int& colSize);
(This function holds the main functionality of the 3d string matching)

void DeleteArray(char***& arr, int x, int y, int z);
(This function deletes the dynamically created array)

Note: string library not allowed.

Q3: Recursive Functions

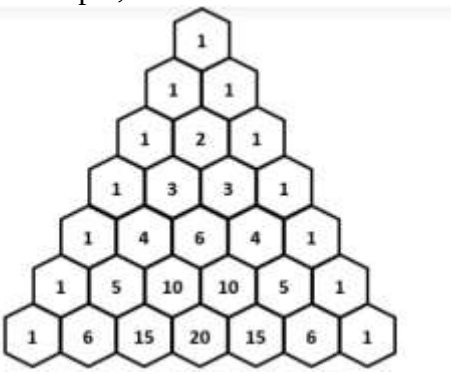
Marks: 10

1. Given a number n, write a recursive function whether it is a perfect number or not. Your function should return true or false. A perfect number is a positive integer that is equal to the sum of its proper divisors.
   Example:
   6 is a perfect number, which is a sum of 1, 2 and 3 which are the divisors of 6.
   Function Prototype: *bool isperfectNumber(int n);*

   Marks: 10
2. Given a string, write a recursive function to find and return the total number of vowels.
   Function Prototype: *int findVowels(char* str);*

   Marks: 10
3. Write a C++ recursive program to compute the value of a given position in Pascal's Triangle. Your function should return the value at that index in Pascal's triangle. The entire implementation must be done recursively.
   Function Prototype: *int pascal(int row,int col);*

   For example,



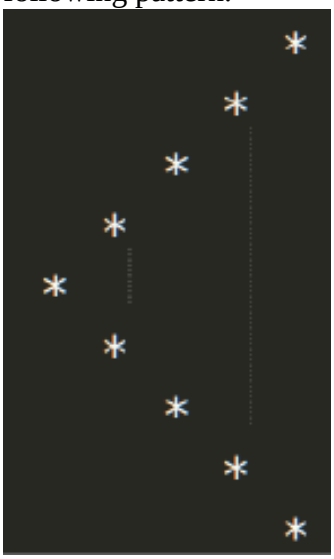
Input : Row = 2, Col = 1
Output : 2
Input : Row = 4, Col = 2
Output : 6

Marks: 15

4. Write a C++ recursive function to print following pattern. No loops allowed whatsoever, and you can write other helping (recursive) functions. Your **function prototype** must be as follows:

   void PrintPattern1(int start, int end);

   For example, calling your function with these argument PrintPattern1(1,10) should print following pattern.

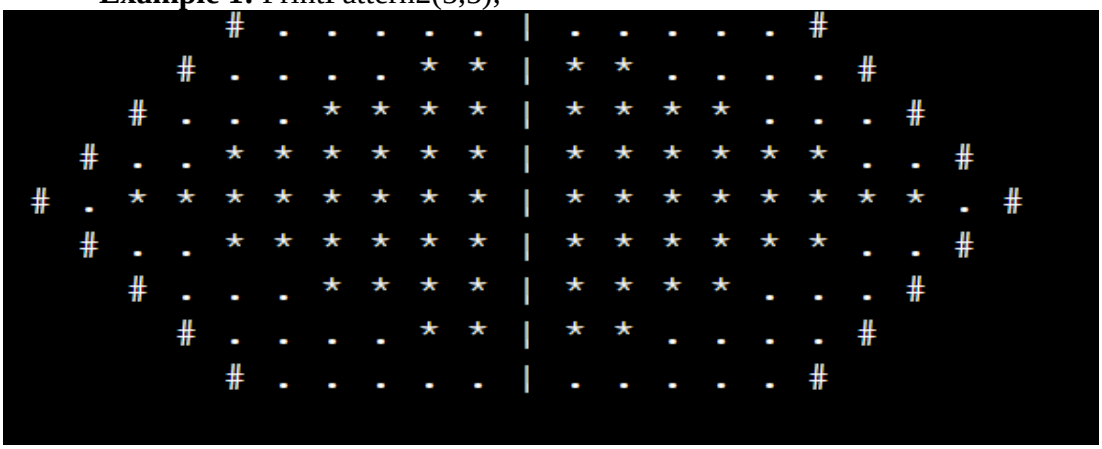


Marks: 20

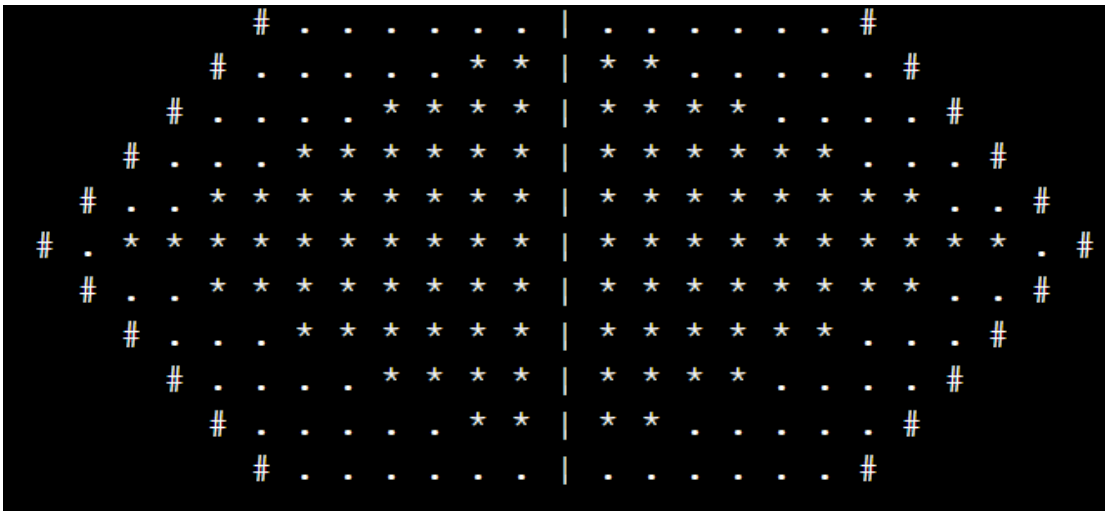
5. Write a C++ code to recursively print a hollow diamond structure. No loops are allowed whatsoever, and you can use helper functions. The **function prototype** is as follows:
   void printHollowDiamond(int n);
   Here n is the number of stars.
   Example: *printHollowDiamond(5)* will give us the following output

Marks: 15

6. Write a C++ recursive function PrintPattern2 to print following pattern using recursion. No loops allowed whatsoever, and you can write other helping (recursive) functions (other than main). Your **function prototype** must be as follows:
   void PrintPattern2(int , int);
   Example 1: PrintPattern2(5,5);



Example 2:
PrintPattern2(6,6);



Marks: 50

Q4: Snake Game

Figure 1: Snake board game

You are required to do the following:

- Create a snake board of size M rows and N columns (M and N can be any numbers hence the grid can be of dynamic size). As C++ is a row major language, game will start at [M-1][1] and end at [0][N-1] (for odd M) or at [0][0] (for even M).
- Randomly generate N-1 snakes on the board. In order to generate snake you only need to know head and tail of the snake. Make sure that both head and tail are on the board. Moreover, if head is on row ($\mathbf{M}_i$)and tail is on row ($\mathbf{M}_j$) then i will always be less than j.
- Similarly, generate N-1 ladders on the board.
- In order to start the player, need a six on the dice. Once the game is started display the output on the dice and wait for key press before second player's turn.
- Game will go on until one player wins the game.
- In case player lands on a snake's head, it will come down to its tail (***use aesthetic sense***), here you need to display a message "oops, snake got you!!!"

- In case player lands on the bottom of the leader it will climb the ladder, here you need to display a message “you got lucky”.
- **The controller function is a must with the same prototype as described below, you can make other helper functions based on your own understandings.**

Function Prototype:

void startSnakeGame(); // controller of the game

some helper functions:

1. snakeBoardCreation
2. displayBoard
3. playdice
4. checkSnakeHead
5. checkLeaderFoot … etc

Note: global variable not allowed, and you can create other helping functions.

Happy Coding 😊
