

Doctorado en Ciencias en Computación
Maestría en Ciencias en Ciencias de la Computación

Problemario de Programación

1. Se propone un conjunto de problemas con nivel de dificultad 1, 2 y 3.
2. El examen incluye la descripción del problema y la entrada y salida que será utilizada por el aspirante para probar el funcionamiento de cada programa.
3. La evaluación de los problemas es binaria, el programa funciona correctamente o no. Esta evaluación se realiza usando una entrada y salida que no será conocida por el aspirante.
4. El examen consiste de uno o varios problemas de cada uno de los niveles de dificultad y tendrá una duración de 4 horas.
5. La calificación del examen depende del número y del tipo de problema resuelto correctamente y se determina con base en la siguiente tabla:

Problemas resueltos	Calificación
Uno de nivel 1 y uno de nivel 2	70
Uno de nivel 3	80
Uno de nivel 3 y uno de nivel 1	90
Uno de nivel 3 y uno de nivel 2	100

6. Con el propósito de que los aspirantes puedan corregir los problemas cuyos programas no funcionen correctamente, los problemas serán evaluados conforme los vayan terminando y entregando. Si un programa no funciona correctamente se le pueden hacer al aspirante las indicaciones pertinentes:
 - a. El ejecutable no se puede abrir
 - b. No lee del directorio actual
 - c. No guarda en el directorio actual
 - d. No produce correctamente la salida
7. Los problemas de nivel de dificultad 1 prácticamente no requieren análisis y solo se necesita codificar un algoritmo muy simple utilizando estructuras de control y de datos básicas (vectores y arreglos). Se consideran problemas de este tipo los siguientes (y similares):
 - a. Evaluación de funciones que involucran el cálculo de series.
 - b. Solución de ecuaciones cuadráticas.
 - c. Operaciones con matrices cuadradas.
 - d. Cálculo de determinantes de matrices 2×2 y 3×3 .
8. Los problemas de nivel de dificultad 2 prácticamente no requieren análisis y solo se necesita codificar un algoritmo que utiliza estructuras de control básicas y estructuras de datos tales como arreglos, pilas y colas. Se consideran problemas de este tipo los siguientes (y similares):
 - a. Dado un grafo, determinar si un conjunto de vértices es un clique, o una cubierta.
 - b. Dado un grafo, realizar un recorrido en anchura (profundidad) a partir de un vértice dado.

9. Los problemas de nivel de dificultad 3 requieren análisis y se necesita diseñar y codificar un algoritmo que utiliza estructuras de control básicas y estructuras de datos tales como arreglos, pilas y colas. Estos problemas son del tipo concurso ACM.

NOTAS IMPORTANTES

1. En todos los casos **se requiere que la entrada y salida se realice usando archivos de texto ubicados el directorio actual**. En la especificación de cada problema se muestra un ejemplo de entrada y salida para que se pruebe el funcionamiento del programa.
2. En todos los problemas se solicita al aspirante que realice un programa con un lenguaje de programación de propósito general y que **entregue una descripción del algoritmo usado en formato libre, su código fuente y el archivo ejecutable**.
3. La descripción del algoritmo se requiere para tener la posibilidad de evaluar manualmente un problema de nivel de dificultad 3. Los problemas de dificultad menor solo se evalúan en forma binaria (funcionan o no).

Problema con nivel de dificultad 1.

1. Dada la función $f(n) = \sum_{i=1}^n i^2$.

Escriba un programa que dado n , determine $f(n)$.

Entrada: Archivo de texto **in.txt** que contiene una lista de valores de n , escritos uno en cada línea.

in.txt

Salida. Archivo de texto **out.txt** que contiene la lista de valores $f(n)$ correspondientes a los valores de la entrada, escritos uno en cada línea.

Pruebe su programa con la entrada y salida siguientes:

in.txt

1

2

3

5

out.txt

1

5

14

55

Problema con nivel de dificultad 2.

Dado un grafo $G=(V, E)$, un conjunto de vértices $S \subseteq V$ se dice que es un cliqué, si y solo si todos los vértices de S están conectados entre si.

Escriba un programa que dado un conjunto de vértices S , determine si es o no un cliqué.

Entrada: Archivo de texto **in.txt** que contiene el número de vértices del grafo, la matriz de adyacencia del grafo G y los conjuntos de vértices S_1, S_2, \dots, S_k .

Cada S_i define los vértices con respecto a los cuales se debe determinar si son o no un cliqué.

Salida. Archivo de texto **out.txt** que contiene las respuestas correspondientes a cada conjunto de vértices (si ó no es un cliqué).

Pruebe su programa con la entrada y salida siguientes:

in.txt

4

0 0 1 1

1 0 0 1

1 1 0 1

1 1 1 0

1 2

3 4

1 3 4

1 2 4

out.txt

no

si

si

no

Problema con nivel de dificultad 3.

14 sep 2001

ACM Local Programming Contest 2001 Problem #3. Spiral

Program: PROG3.EXE, PROG3.CLASS
Source Program: PROG3.CPP, PROG3.C
PROG3.JAVA, PROG3.PAS
Input file: PROG3.IN Output file: PROG3.OUT

Problem Description

Write de program that reads a matrix of size $n \times n$ and order it from smaller to larger in clockwise spiral form. The spiral always begins in a given corner.

Input

In the first line an integer representing the n value ($1 \leq n \leq 100$)

In the second line two integers representing the initial corner coordinates.

In the n following lines, the n elements of each matrix row separated by white spaces

Output

The ordered matrix in n rows of n elements

Input file

```
3  
0 0  
7 4 13  
10 15 20  
5 7 2
```

Output file

```
2 4 5  
15 20 7  
13 10 7
```