



UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Ciencias

Escuela Profesional de Ciencia de la Computación

EXAMEN PARCIAL

Curso: **Fundamentos de Programación CC112 - A**

Nombres y Apellidos: Victor Hugo Cadillo Gutierrez

Código: 20255514D

1.a.

```
[victor@victor-archdesktop parcial]$ make debug
mkdir -p build/debug
g++ -std=c++17 -Wall -Wextra -Wpedantic -g -O0 prob01.cpp -o build/debug/prob01
[victor@victor-archdesktop parcial]$ ./build/debug/prob01
1 2 3 4 5
Costo maximo: 121
Salidas de la recursion
Llamada en rama: 4 costo en ejecucion recursiva: 5
Llamada en rama: 3 costo en ejecucion recursiva: 9
Llamada en rama: 3 costo en ejecucion recursiva: 20
Llamada en rama: 2 costo en ejecucion recursiva: 23
Llamada en rama: 3 costo en ejecucion recursiva: 5
Llamada en rama: 2 costo en ejecucion recursiva: 17
Llamada en rama: 2 costo en ejecucion recursiva: 60
Llamada en rama: 1 costo en ejecucion recursiva: 62
Llamada en rama: 3 costo en ejecucion recursiva: 5
Llamada en rama: 2 costo en ejecucion recursiva: 9
Llamada en rama: 2 costo en ejecucion recursiva: 20
Llamada en rama: 1 costo en ejecucion recursiva: 26
Llamada en rama: 2 costo en ejecucion recursiva: 5
```

1.b.

El algoritmo recursivo se llama desde un bucle for por lo que se ramifica en varias opciones que luego compara con el costo total para encontrar el valor máximo.

```

Llamada en rama: 1 costo en ejecucion recursiva: 26
Llamada en rama: 2 costo en ejecucion recursiva: 5
Llamada en rama: 1 costo en ejecucion recursiva: 29
Llamada en rama: 1 costo en ejecucion recursiva: 120
Llamada en rama: 0 costo en ejecucion recursiva: 121
Llamada en rama: 3 costo en ejecucion recursiva: 5
Llamada en rama: 2 costo en ejecucion recursiva: 9
Llamada en rama: 2 costo en ejecucion recursiva: 20
Llamada en rama: 1 costo en ejecucion recursiva: 23
Llamada en rama: 2 costo en ejecucion recursiva: 5
Llamada en rama: 1 costo en ejecucion recursiva: 17
Llamada en rama: 1 costo en ejecucion recursiva: 60
Llamada en rama: 0 costo en ejecucion recursiva: 62
Llamada en rama: 2 costo en ejecucion recursiva: 5
Llamada en rama: 1 costo en ejecucion recursiva: 9
Llamada en rama: 1 costo en ejecucion recursiva: 20
Llamada en rama: 0 costo en ejecucion recursiva: 26
Llamada en rama: 1 costo en ejecucion recursiva: 5
Llamada en rama: 0 costo en ejecucion recursiva: 29
Llamada en rama: 0 costo en ejecucion recursiva: 120

```

2.

```

[victor@victor-archdesktop parcial]$ make debug
mkdir -p build/debug
g++ -std=c++17 -Wall -Wextra -Wpedantic -g -O0 prob02.cpp -o build/debug/prob02
[victor@victor-archdesktop parcial]$ ./build/debug/prob02
Ingrese numero de corredores: 5
Nombre del corredor 1: Angel
Tiempo (en segundos): 360
Nombre del corredor 2: Luis
Tiempo (en segundos): 420
Nombre del corredor 3: Maria
Tiempo (en segundos): 390
Nombre del corredor 4: Jorge
Tiempo (en segundos): 360
Nombre del corredor 5: Carlos
Tiempo (en segundos): 450

RANKING FINAL
1. Angel - 360 segundos

```

RANKING FINAL

1. Angel - 360 segundos
2. Jorge - 360 segundos
3. Maria - 390 segundos
4. Luis - 420 segundos
5. Carlos - 450 segundos

Ingrese el nombre a buscar: Luis

Corredor encontrado

Posicion: 4

Tiempo: 420 segundos

Ingrese el tiempo minimo y maximo: 350 420

Corredores en el rango [350, 420]:

Angel - 360 segundos

Jorge - 360 segundos

Maria - 390 segundos

Luis - 420 segundos

3.

```
[victor@victor-archdesktop parcial]$ make debug
mkdir -p build/debug
g++ -std=c++17 -Wall -Wextra -Wpedantic -g -O0 prob03.cpp -o build/debug/prob03
[victor@victor-archdesktop parcial]$ ./build/debug/prob03
Entrada:
2 3 1 4
1 2 3 2
5 6 7 8
0 1 2 1

Salida:
Fila          Dominante Estricta          Fuertemente Dominante
-----
0              NO              NO
1              NO              NO
2              SI              SI
3              NO              NO
```

4.

```
[victor@victor-archdesktop parcial]$ make debug
mkdir -p build/debug
g++ -std=c++17 -Wall -Wextra -Wpedantic -g -O0 prob04.cpp -o build/debug/prob04
[victor@victor-archdesktop parcial]$ valgrind ./build/debug/prob04
==86146== Memcheck, a memory error detector
==86146== Copyright (C) 2002-2024, and GNU GPL'd, by Julian Seward et al.
==86146== Using Valgrind-3.25.1 and LibVEX; rerun with -h for copyright info
==86146== Command: ./build/debug/prob04
==86146==
Ingrese el tamaño del arreglo: 4
Ingrese la cantidad de estudiantes a registrar: 3
Codigo : 101
Promedio : 14.5
Codigo : 102
Promedio : 8.
Codigo : 103
Promedio : 16.2
Codigo: 101 Promedio: 14.5
Codigo: 102 Promedio: 8
Codigo: 103 Promedio: 16.2
```

```
Codigo: 102 Promedio: 8
Codigo: 103 Promedio: 16.2
Ingrese la cantidad de estudiantes que quiere agregar: 2
Codigo : 104
Promedio : 9.5
Codigo : 105
Promedio : 12.3
Estudiantes registrados:
Codigo: 101 Promedio: 14.5
Codigo: 102 Promedio: 8
Codigo: 103 Promedio: 16.2
Codigo: 104 Promedio: 9.5
Codigo: 105 Promedio: 12.3

Filtrando a estudiantes desaprobados...

Estudiantes aprobados:
Codigo: 101 Promedio: 14.5
```

```
Estudiantes aprobados:
Codigo: 101 Promedio: 14.5
Codigo: 103 Promedio: 16.2
Codigo: 105 Promedio: 12.3
=====
Informacion adicional para monitorear como cambian los tamanios del array.
Tamanio del array inicial: 4
Tamanio del array con los datos agregados: 8
Tamanio del array con los datos filtrados: 3
==86146==
==86146== HEAP SUMMARY:
==86146==    in use at exit: 0 bytes in 0 blocks
==86146==   total heap usage: 14 allocs, 14 frees, 76,181 bytes allocated
==86146==
==86146== All heap blocks were freed -- no leaks are possible
==86146==
==86146== For lists of detected and suppressed errors, rerun with: -s
==86146== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```