



Examen Sustitutorio 15/07/2019

CC112

15/7/2019

Ciclo: 2019-1

Normas:

1. El alumno entregará esta hoja de examen debidamente llenada con sus datos.
2. La solución de la prueba se guardarán en **Escritorio**, carpeta: **ApellidoNombreCodigo** (sin espacios en blanco), la pregunta **n** se guardará en el archivo: **n.c** ($n = 1, 2, \dots$).
3. No se permite: El uso de celulares, internet, USB, ingresar después de 15 min. de iniciado el examen ni salir antes de la hora de finalización.
4. Todo acto anti-ético será amonestado y registrado en el historial del estudiante.

Apellidos : _____ Nombres : _____

Sección : _____ Grupo: _____

1. [5 ptos.] Escriba un programa que defina:

int n=5, arr[n], *p=arr...

UTILICE APUNTADES (obligatorio) para todo el resto del programa que hará lo siguiente:

Lee del teclado 5 enteros y los asigna a elementos de **arr** en modo sucesivo, luego reporta el índice, la dirección en la RAM y el valor asignado. Finalmente reporta el máximo de los números ingresados. Su salida puede ser:

Ingrese 5 enteros:

Numero (1): 5

Numero (2): 4

Numero (3): 3

Numero (4): 2

Numero (5): 1

Valores del arreglo

índice	Dirección	Valor
0	0x7fff852a0ee0	5
1	0x7fff852a0ee4	4
2	0x7fff852a0ee8	3
3	0x7fff852a0eec	2
4	0x7fff852a0ef0	1

El maximo es: 5

2. [5 ptos.] Escriba un programa que lea 5 frases cortas (≤ 30 caracteres), luego llame a una función para ubicar las 5 vocales (a,e,i,o,u) en cada frase. Su salida puede ser:

[CADENA DE CARACTERES]

Ubicar las vocales

Ingrese palabra 1: si

Ingrese palabra 2: no

Ingrese palabra 3: si no

Ingrese palabra 4: noo

Ingrese palabra 5: hmm

En la frase: 'si' se ubico:

Vocal i en posicion:2

En la frase: 'no' se ubico:

Vocal o en posicion:2

En la frase: 'si no' se ubico:

Vocal i en posicion:2

Vocal o en posicion:5

En la frase: 'noo' se ubico:

Vocal o en posicion:2

Vocal o en posicion:3

En la frase: 'hmm' se ubico:

Sugerencia: Utilice **cin** y **cout** y lea una frase con: `cin.getline(frase, 31)`

3. [5 pts.] Una matriz **poco densa** es una matriz con una muy alta cantidad de ceros entre sus elementos. Por ejemplo, la matriz A de 500x700 solo tiene datos:

1	0	0	0
0	2	0	0
0	0	0	0
0	0	0	0

Para ahorrar espacio, se guarda cada entrada diferente de cero en un **vector** de enteros en **memoria dinámica**, para cada elemento se guarde 3 datos (**fila, columna, valor**):

0	0	1	1	1	2
0	1	2	3	4	5

Escriba un programa que llame a una función para llenar **dinámicamente** a **vector**, tome como entrada **la fila, la columna y el valor** de cada uno de los elementos de la matriz poco densa. Luego llame a otra función para listar los elementos de la matriz. Un ejemplo de salida es:

```

fila=0
columna= 0
valor= 1
fila= 1
columna= 1
valor= 2
fila= 0
columna= 0
valor= 0
i j Valor
0 0 1
1 1 2

```

4. [5 pts.] Programe un diccionario de **n** palabras con longitud máxima 15 caracteres. Para ello, lea **n>0**, cree un area dinámica para alojar n palabras, luego lea las n palabras del teclado y finalmente use el diccionario: lea palabras e indique si está o no, en el diccionario; termine cuando ingrese enter al teclado. Su salida puede ser:

```

Cuántas palabras tendrá el diccionario?: 2
Palabra 1: aa
Palabra 2: dd
Qué palabra busco? (pulsa retorno para acabar): aa
Sí está.
Qué palabra busco? (pulsa retorno para acabar): dd
Sí está.
Qué palabra busco? (pulsa retorno para acabar): ee
No está.
Qué palabra busco? (pulsa retorno para acabar):

```

Sugerencia:

Utilice cin, cout para la entrada y salida de datos:

```

#include<iostream>
using namespace std;

```

Defina : char linea[16];

Lea una línea con: cin.getline(linea, 16)

Atento: lea n así:

```

cin >> n; // lee n
while(getchar()!=10); // vacia el buffer inmediatamente.

```