Universidad Nacional de Ingeniería
Facultad de Ciencias
Escuela Profesional de Ciencia de la Computación
Fundamentos de programación

Examen Sustitutorio 15/07/2019

CC112 **15/7/2019** **Ciclo: 2019-1**

1. [5 ptos.] Escriba un programa que defina:
   int n=5, **arr**[n], *p=arr...
   UTILICE APUNTADORES (obligatorio) para todo el resto del programa que hará lo siguiente:
   Lee del teclado 5 enteros y los asigana 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
indice  Dirección       Valor
  0     0x7fff852a0ee0   5
  1     0x7fff852a0ee4   4
  2     0x7fff852a0ee8   3
  3     0x7fff852a0eec   2
  4     0x7fff852a0ef0   1
El maximo es: 5
```

```
//1.c
#include <iostream>
using namespace std;
main(){
        int n=5, arr[n], *p=arr, i, max;
        cout << "\nIngrese " << n << " enteros:\n";
        for (i=0; i < n; i++){
                cout << "Numero ("<< i+1 <<"): ";
                cin >> *p++;
        }
        cout << "\nValores del arreglo" << endl;
        cout << "indice Dirección      Valor"  << endl;
        for (i=0, p=arr, max = *p; i<n; i++){
                if(max<*p) max = *p;
                cout << "  " << i << "    " << p << "  " << *p++ << endl;
        }
        cout << "\nEl maximo es: " << max << endl;
}
```

2. [5 ptos.] Escriba un programa que lea 5 frasee 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)

```
//2.c
#include<iostream>
using namespace std;
# define NL 31
void ubicar_vocales(char cad[]){
	int i=0;
	while(cad[i]){
		if(cad[i]=='a'||cad[i]=='A') cout<<" Vocal a en posicion:"<<i+1 << endl;
		else if(cad[i]=='e'||cad[i]=='E') cout<<" Vocal e en posicion:"<<i+1 << endl;
			else if(cad[i]=='i'||cad[i]=='I') cout<<" Vocal i en posicion:"<<i+1 << endl;
				else if(cad[i]=='o'||cad[i]=='O') cout<<" Vocal o en posicion:"<<i+1 << endl;
					else if(cad[i]=='u'||cad[i]=='U') cout<<" Vocal u en posicion:"<<i+1 << endl;
		i++;
	}
}
main(){
	int i, n=5;
	char cad[n][NL];
	cout<<"\n\t[  CADENA DE CARACTERES  ]\n";
	cout<<" Ubicar las vocales " << endl;
	for(i=0; i<n; i++){
		cout<<" Ingrese palabra " << i+1 << ": ";
		cin.getline(cad[i], NL);
	}
	for(i=0; i<n; i++){
		cout<<"\nEn la frase: '"<<cad[i]<<"' se ubico:"<<endl;
		ubicar_vocales(cad[i]);
	}
}
```

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 **<u>memoria dinámica</u>**, 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 **<u>dinámicamente</u>** 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

```
//3.c
#include <iostream>
#include <cstdio>
#include <cstdlib>
using namespace std;
int *insertarDato(int f, int c, int d, int *v, int *n);
void mostrarVector(int *v, int n);
int main(){
   int f,c,d,n=0;
   int *v = NULL;
   do{
                        cout<<"\nfila= "; cin>>f;
                        cout<<"columna= "; cin>>c;
                        cout<<"valor= "; cin>>d;
                        if (d!=0) v = insertarDato(f,c,d,v,&n);
           } while (d!=0);
           mostrarVector(v,n);
           free(v);
}
int *insertarDato(int f, int c, int d, int *v, int *n){
           v=(int* )realloc( v,(*n+1)*sizeof(int)*3 );
           *(v+*n)=f;
           *(v+*n+1)=c;
           *(v+*n+2)=d;
           *n +=3;
           return v;
}
void mostrarVector(int *v, int n){
           int i=0;
           cout << "\ni  j  Valor\n";
           for(i = 0; i < n; i +=3) cout << *v++ << "  " << *v++ << "  " << *v++ << endl;
}
```

4. [5 ptos.] Programe un diccionario de **n** palabras con longitud náxima 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.

```
// 4.c
#include <iostream>
//#include <cstdlib>
#include <cstring>
using namespace std;
#define MAXLON 16
int buscar(int n, char *d, char *pal){
        int i;
        for(i=0; i<n; i++, d +=MAXLON) if (strcmp(d, pal)==0) return 1;
        return 0;
}
main() {
        char *diccionario, *d, *linea;
        int i=0, n;
        cout << "Cuántas palabras tendrá el diccionario?: ";
        cin >> n;
        while(getchar()!=10);           // vacia el buffer
        d = diccionario = (char *)malloc((n*MAXLON)*sizeof(char));
        while (i < n) {   // lee las palabras
                cout << "Palabra " << i+1 << ": ";
                cin.getline(linea, MAXLON);
                strcpy(d, linea);
                d += MAXLON;
                i++;
        }
        do {                                                                    // usa el diccionario
                printf ("Qué palabra busco? (pulsa retorno para acabar): ");
                cin.getline(linea, MAXLON);
                if (strlen(linea) == 0) break;
                if (buscar(n, diccionario, linea)) printf ("Sí está.\n");
                else printf ("No está.\n");
        } while(1);
        free(diccionario);
}
```