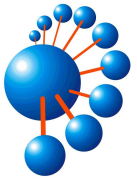
Programación de Computadores
2023-2

Repaso

Universidad de Concepción

José Fuentes - jfuentess@inf.udec.cl

Departamento de Ingeniería Informática y Ciencias de la Computación



Ejercicio 1: Ciclo **while** a ciclo **for**

```c
// ¿Cómo reemplazamos el ciclo while por un ciclo for?
int strlen_local(char *t) {
  int largo = 0;

  while(*t != '\0') {
    largo ++;
    t++;
  }
  return largo;
}

int main() {
  char texto[128];

  printf("Ingrese un texto: ");
  scanf("%[^\n]s", texto);

  int l = strlen_local(texto);
  printf("El largo del texto ingresado es %d\n", l);

  return 1;
}
```

Ver: while2for.c

Ejercicio 2: Ciclo **for** a ciclo **while**

```c
// ¿Cómo cambiamos el ciclo for de esta función por un ciclo while?

// Función verifica si el elemento v está presente en el arreglo
// ordenado arr de largo n
int busquedaBinaria(int n, int arr[], int v) {
  for(int l=0, r=n-1; l<=r;) {

    int m = (l + r)/2;

    if (arr[m] == v)
      return m;

    if (arr[m] < v)
      l = m + 1;

    else
      r = m - 1;
  }

  return -1;
}
```

Ver: for2while.c

Ejercicio 3: Seguimiento

```c
// ¿Qué hace la siguiente implementación?
void func(int a, int b, int *c, int *d) {
  int x=0;
  while(a >= b) {
    x++;
    a -= b;
  }
  *c = x;
  *d = a;
}

int main(void) {
  int a, b, c=0, r=0;
  printf("Ingrese dos valores enteros: ");
  scanf("%d %d", &a, &b);

  func(a, b, &c, &r);
  printf("c: %d, r: %d\n", c, r);

  return EXIT_SUCCESS;
}
```

Ver: seguimiento.c

Ejercicio 4: Arreglos

```c
#include <stdio.h>
#include <stdlib.h>

int main(void) {
  int n;
  printf("Ingrese el largo del arreglo: ");
  scanf("%d", &n);

  int x[n];
  for(int i=0; i < n; i++) x[i] = rand() % 100;

  for(int i=0; i < n; i++) printf("%d ", x[i]);
  printf("\n");

  /* Escriba un programa que, dado un arreglo de enteros x de largo n, genere dos
     arreglos x1 y x2, donde x1 contiene todos los valores pares del arreglo x y
     x2 contiene todos los valores impares. */

  return EXIT_SUCCESS;
}
```

Ver: pares_impares.c

Ejercicio 5: Matrices

```c
int main(void) {
  int n, m;
  printf("Ingrese las dimensiones de la matriz: ");
  scanf("%d %d", &n, &m);

  int M[n][m];

  for(int i=0; i < n; i++)
    for(int j=0; j < m; j++)
      // Con probabilidad del 50% las celdas tendrás valores válidos entre 0
      // y 99. Las otras celdas quedarán sin valor, representadas por un valor -1
      if(rand() % 2 == 1) M[i][j] = rand() % 100;
      else M[i][j] = -1;

  /* Complete TODAS las celdas con valor -1 utilizando las reglas:
       - Si una celda con valor -1 está rodeada sólo por celdas con valores -1, entonces no
         cambiará de valor
       - Si una celda con valor -1 está rodeada por al menos una celda con valor distinto a -1,
         entonces su nuevo valor será el promedio de las celdas vecinas con valores distintos a -1 */
  return EXIT_SUCCESS;
}
```

Ver: propagacion.c

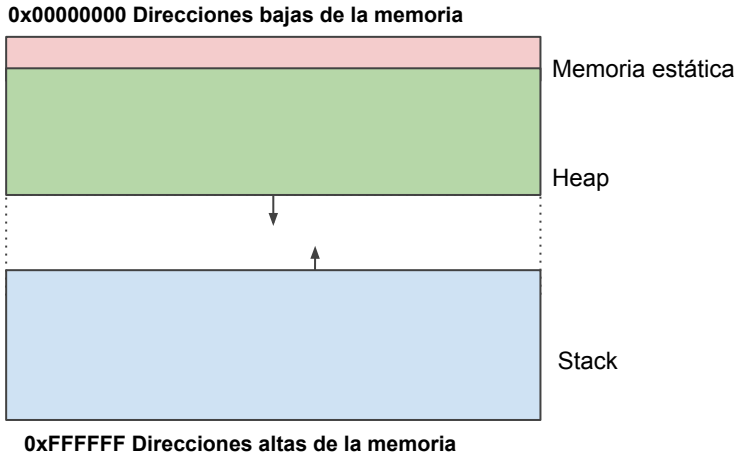
Estadísticas de un arreglo

Realizar seguimiento

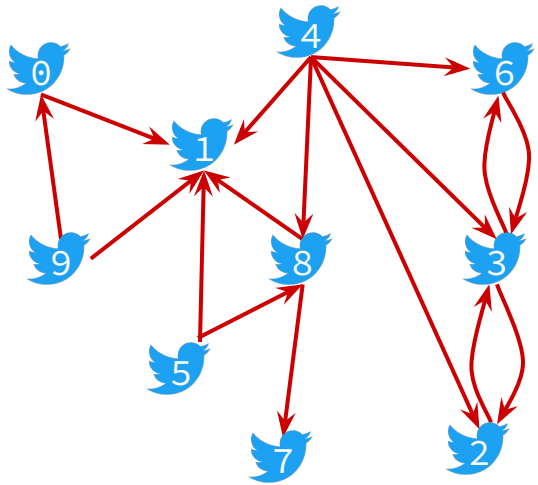
```c
#include <stdlib.h>
#include <stdio.h>

void estadisticas(int n, int x[n], int *m,
                  int *M, float *p) {
  int min = x[0], max = x[0];
  int prom = x[0];
  for(int i=1; i < n; i++) {
    prom += x[i];
    if(x[i] < min)
      min = x[i];
    else if(x[i] > max)
      max = x[i];
  }

  *m = min;
  *M = max;
  *p = (float)prom / n;
}
```

Ver: estadisticas.c

```c
int main() {
  int n;
  printf("Ingrese el largo del arreglo: ");
  scanf("%d", &n);

  int x[n]; // ¿cómo cambiamos a memoria dinámica (malloc)?
  for(int i=0; i < n; i++) x[i] = rand() % 100;

  int mi=0, ma=0;
  float pr = 0;

  estadisticas(n, x, &mi, &ma, &pr);

  printf("Mínimo: %d, máximo: %d, promedio: %.2f \n", mi, ma, pr);
  return EXIT_SUCCESS;
}
```



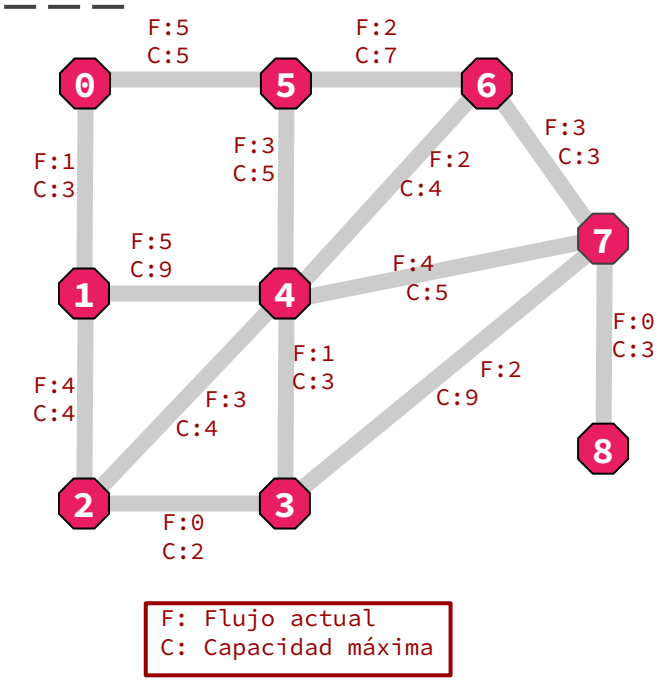
Uso de matrices: redes sociales



	0	1	2	3	4	5	6	7	8	9
0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	1	0	0	0	1	0	0	0
4	0	1	1	1	0	0	1	0	1	0
5	0	1	0	0	0	0	0	0	1	0
6	0	0	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	1	0	0
9	1	1	0	0	0	0	0	0	0	0

Ver: red_social.c

Uso de matrices: redes de tuberías



	0	1	2	3	4	5	6	7	8
0	– –	1 3	– –	– –	– –	5 5	– –	– –	– –
1	1 3	– –	4 4	– –	5 9	– –	– –	– –	– –
2	– –	4 4	– –	0 2	3 4	– –	– –	– –	– –
3	– –	– –	0 2	– –	1 3	– –	– –	2 9	– –
4	– –	5 9	3 4	1 3		3 5	2 4	4 5	– –
5	5 5	– –	– –	– –	3 5	– –	2 7	– –	– –
6	– –	– –	– –	– –	2 4	2 7	– –	3 3	– –
7	– –	– –			2 9	4 5	– –	3 3	– –	0 3
8	– –	– –	– –	– –	– –	– –	– –	0 3	– –

F C

Ver: tuberias.c