Examen de Programación 1 Grado en Ingeniería Informática
11 de Enero de 2016

Nombre:

NIF:

ATENCIÓN: Responded los problemas 1, 2 y 3 en las hojas de examen. Justificad brevemente la respuesta.

Problema 1 (2 puntos)

Indicad si el siguiente programa es correcto. Si no es correcto indicad y corregid los errores. En caso contrario, indicad **los valores mostrados por pantalla**. Justificad brevemente la respuesta.

```
#include <stdio.h>
// Prototipos de las acciones y funciones auxiliares
void B(int &x, int &y);
int A(int x, int y);
int main( ){
  int a[2] = {5, 6};
  int c = 1;
  B(a[0], a[1]);
  printf("Main:\n a[0] = %d \t a[1] = %d \t c = %d \n", a[0], a[1], c);
}
// Acciones y funciones auxiliares
void B(int &x, int &y){
  int c;
  c = A(x, y);
  x=8;
  y=10;
  printf("B:\n c = %d \n",c);
}
int A(int x, int y){
  x = 2*x;
  y = 2*y;
  printf("A:\n x = %d \t y = %d \n", x, y);
  return(x-y);
}
```

Problema 2 (2 puntos)

Indicad si el siguiente programa es correcto. Si no es correcto indicad y corregid los errores. En caso contrario, indicad **los valores mostrados por pantalla**. Justificad brevemente la respuesta.

```
#include<stdio.h>
#define M 5
// Prototipos de las acciones auxiliares
void transformar(int m1[M][M], int m2[M][M], int n);
void mostrar(int m[M][M], int n);
int main( ){
   int m1[M][M] = {{1,1,1,1,1}, {2,2,2,2,2}, {3,3,3,3,3}, {4,4,4,4,4}, {5,5,5,5,5}};
   int m2[M][M];
   transformar(m1, m2, M);
   mostrar(m1, M-2);
   mostrar(m2, M-1);
}
// Acciones auxiliares
void transformar(int m1[M][M], int m2[M][M], int n){
  int x, y;
  for (x = 0; x < n; x++){              // límite n
        for (y = 0; y < n; y++){        // límite n
                m2[y][x] = m1[x][y];
        }
   }
}
void mostrar(int m[M][M], int n){
  int i, j;
  for (i = 0; i < n; i++){              // límite n
       for (j = i; j < n; j++){         // inicio en i, límite n
            printf("%d \t", m[i][j]);
        }
        putchar('\n');
   }
}
```

Nombre:

NIF:

Problema 3 (2 puntos)

Indicad si el siguiente programa es correcto. Si no es correcto indicad y corregid los errores. En caso contrario, indicad **los valores mostrados por pantalla**. Justificad brevemente la respuesta.

```
#include<stdio.h>
#define N 5
#define MAX 20
// Prototipo de la  función auxiliar
char relacion(char p1[MAX+1] , char p2[MAX+1]);
int main( ){
   char p[N][MAX+1] = {"abcd", "abc", "abcde",  "abfg", "abcd"};
   int i, j;
   char c;
   for (i = 0; i < N; i++)
        for (j = N-1; j > i; j--){
              c = relacion(p[i], p[j]);
              printf("Palabra %s \t Palabra %s \t Caracter %c \n", p[i], p[j], c );
        }
}
// Función auxiliar
char relacion(char p1[MAX+1] , char p2[MAX+1]){
  int i;
  bool ok;
  i=0;
  ok= true;
  while (ok && p1[i] != '\0' && p2[i] != '\0'){
        if (p1[i] != p2[i]){
            ok=false;
        } else{
             i=i+1;
        }
   } // final del while
   if  (p1[i] != '\0') {
        return(p1[i]);
    } else if (p2[i] != '\0'){
                  return(p2[i]);
               } else{
                   return ('*');
                  }
}
```

Problema 4 (4 puntos)
Utilizando la técnica del diseño descendente implementad **una función** que llamaremos `subcadena` que tenga los parámetros **de entrada** siguientes:

1. Dos tablas $t1$ y $t2$ de caracteres marcadas con la marca NUL '\0' (dos strings) de longitud máxima 60 más una posición para la marca de final de string.
2. Dos valores enteros a y b.

y devuelva (valor de retorno) la posición de inicio en el string $t1$ que contiene la subcadena (fragmento) del string $t2$ entre las posiciones a y b (subcadena entre $t2[a]$ y $t2[b]$, ambas posiciones incluidas).

IMPORTANTE:

- Considerad las posiciones de los caracteres dentro de los strings $t1$ y $t2$ numerados de 0 a la longitud del correspondiente string menos 1.
- En caso que los caracteres de $t2$ entre les posiciones a y b estén contenidos más de una vez en $t1$, la función `subcadena` devolverá la posición de inicio de la primera ocurrencia.
- En caso que los caracteres de $t2$ entre les posiciones a y b no estén contenidos en $t1$, la función `subcadena` devolverá el valor -1.
- La función `subcadena` deberá verificar que los parámetros a y b son correctos (dentro del rango) en base a la longitud del string $t2$ (número de caracteres almacenados en $t2$). En caso que los valores de a y b no sean correctos (estén fuera de rango) la función `subcadena` devolverá el valor -1.
- Todas les funciones y acciones auxiliares que necesitéis las debéis implementar, es decir, **no podéis considerar las funciones y acciones definidas en `<string.h>`**.

A modo de ejemplo de ejecución del programa resultante, las llamadas a la función `subcadena` siguientes:

```
#include <stdio.h>
#define M 60
int main( ){
  char t1[M+1] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ-ABC-ABC";
  char t2[M+1] = "DEFGHIJKLMNOP-ABC";
  int n;

  n=subcadena(t1, t2, 3, 6);
  printf("Posición: %d \n",  n);

  n=subcadena(t1, t2, 0, 2);
  printf("Posición: %d \n",  n);

  n=subcadena(t1, t2, 1, 1);
  printf("Posición: %d \n",  n);

  n=subcadena(t1, t2, 14, 16);
  printf("Posición: %d \n",  n);

  n=subcadena(t1, t2, 11, 16);
  printf("Posición: %d \n",  n);

  n=subcadena(t1, t2, 6, 4);
  printf("Posición: %d \n",  n);
```

```
  n=subcadena(t1, t2, 14, 18);
  printf("Posición: %d \n",  n);
}
```

Mostraran los siguientes resultados por pantalla:

```
Posición: 6
Posición: 3
Posición: 4
Posición: 0
Posición: -1
Posición: -1
Posición: -1
```

Explicación de la salida para cada caso:

- La subcadena en $t2$ entre las posiciones 3 y 6 es `GHIJ` y `GHIJ` está contenida en $t1$ a partir de la posición 6.
- La subcadena en $t2$ entre las posiciones 0 y 2 es `DEF` y la subcadena `DEF` está contenida en $t1$ a partir de la posición 3.
- La subcadena en $t2$ entre las posiciones 1 y 1 es `E` y la subcadena `E` está contenida en $t1$ a partir de la posición 4.
- La subcadena en $t2$ entre las posiciones 14 y 16 es `ABC` y la subcadena `ABC` está contenida en $t1$ y la primera ocurrencia está a partir de la posición 0.
- La subcadena en $t2$ entre las posiciones 11 y 16 es `OP-ABC` y la subcadena `OP-ABC` no está contenida en $t1$.
- Los parámetros 6 y 4 son incorrectos (inicio $>$ final).
- Los parámetros 14 y 18 son incorrectos para $t2$ (longitud de $t2$=17, posiciones válidas de caracteres de la 0 a la 16).