IDENTIFICADOR ESTUDIANTE

TALLER DE PROGRAMACIÓN

EXAMEN DEL PRIMER PARCIAL

22 OCTUBRE 2021

ASIENTO

APELLIDOS (en MAYÚSCULAS)

NOMBRE (en MAYÚSCULAS)

PROBLEMA 1 (4 puntos)

Se debe implementar una función que pasándole como parámetros una matriz cuadrada de NxN enteros (siendo N una constante definida previamente), dos valores (f y c) en el rango 0..N-1 que representan una fila y una columna, y un valor ***a*** en el rango 0..N, ponga a 0 las posiciones de la submatriz que empieza en la posición f,c y tiene ***a*** filas y columnas. Por ejemplo, dada la matriz de 5x5 siguiente:

6	**8**	**1**	**4**	**-2**
3	**-12**	**0**	**-14**	**48**
5	**1**	**4**	**16**	**7**
-21	**87**	**-4**	**-8**	**-2**
65	**10**	**2**	**9**	**5**

si se llama a la función pasándole la matriz, 2, 3 (fila 2, columna 3) y 2 (filas y columnas), la matriz debería quedar:

6	**8**	**1**	**4**	**-2**
3	**-12**	**0**	**-14**	**48**
5	**1**	**4**	**0**	**0**
-21	**87**	**-4**	**0**	**0**
65	**10**	**2**	**9**	**5**

□

Se supone que los valores c y f son introducidos correctamente, pero puede que el valor introducido en ***a*** sea incorrecto por no estar en el intervalo [0,N] o que dependiendo de los valores de f o de c, no sea válido. En ambos casos, habría que volver a introducir por teclado dicho valor. Ej.: Con fila=3 y columna=2, a= 3 no es válido porque se "sale" de la matriz.

APARTADO 1 (2 puntos)

Escribir el código que haría falta incluir en el main para leer ***a*** y que en el caso de que ***a*** sea incorrecto o no sea válido, se vuelva a introducir por teclado dicho valor.

APARTADO 2 (2 puntos)

Implementar void **submat_cero** (int mat[N][N], unsigned f, unsigned c, unsigned a).

SOLUCIÓN APARTADO 1:

```
main(){
   int f,c;  int a;
…
// se rellena mat. DEBAJO DE ESTE COMENTARIO NO HAY QUE PONER NADA

// se lee el valor de f y de c. DEBAJO DE ESTE COMENTARIO NO HAY QUE PONER NADA

// comprobar que el valor introducido para a sea correcto y válido
// AQUÍ DEBAJO HAY QUE RESPONDER EL APARTADO 1

do {
   printf("a? "); scanf("%u",&a);
} while(!(a>=0 && a<=N && f+a<=N && c+a<=N));

// se invoca a submat_cero. DEBAJO DE ESTE COMENTARIO NO HAY QUE PONER NADA

…
}
```

SOLUCIÓN APARTADO 2:

```
void submat_cero (int mat[N][N], unsigned f, unsigned c, unsigned a)
{  int i, j;

  for (i=f; i<f+a; i++)
     for (j=c; j<c+a; j++)
        mat[i][j] = 0;
}
```

IDENTIFICADOR ESTUDIANTE

TALLER DE PROGRAMACIÓN

EXAMEN DEL PRIMER PARCIAL

22 OCTUBRE 2021

ASIENTO

APELLIDOS (en MAYÚSCULAS)

NOMBRE (en MAYÚSCULAS)

PROBLEMA 2 (2 puntos)

Se pide implementar una función que dado un array de N=30 caracteres cualesquiera entre la A y la F, indique en otro array de M=6 números enteros, cuántos caracteres de cada tipo contiene el primer array. La función deberá tener la siguiente cabecera:

void contarCaracteres(char* letras, unsigned * cantidades);

Por ejemplo, del siguiente array de caracteres:

[A, B, E, E, B, D, F, A, D, A, F, B, A, F, D, C, F, B, B, F, C, F, C, B, F, D, C, F, B, B]

La función introducirá en el array de enteros:

[4, 8, 4, 4, 2, 8]

NOTA: El array de enteros podría no estar inicializado a 0's cuando se pasa como argumento a la función.

SOLUCIÓN:

```
void contarCaracteres(char * letras, unsigned * cantidades){
        unsigned i;
for (i=0; i<M;i++)
     cantidades[i]=0;
for (i=0;i<N;i++)
     cantidades[letras[i]-'A']++;
 }
```

PROBLEMA 3 (4 puntos)

En una memoria de N=64 bits de ancho de palabra, se rellenan los bits de una palabra (word) de manera aleatoria mediante la función ***void randWord(unsigned word[N])***, se lee por teclado un bit mediante la función ***void scanBit (char * bit)***, buscándose finalmente en palabra *word* cuántos bits coinciden con el bit leido con la función ***unsigned bitsEncontrados(unsigned w[N],char bit)***.

APARTADO 1 (3 puntos)

Definir las funciones *void* ***randWord****(unsigned word[N]), void* ***scanBit*** *(char * bit)* y *unsigned* ***bitsEncontrados****(unsigned word[N],char bit).*

SOLUCIÓN apartado 1:

```
void randWord(unsigned word[N]){
int i;
for (i=0;i<N;i++)
   word[i]=rand()%2;
}

void scanBit (char * bit){
   fflush(stdin);
   scanf("%c",bit);
}

unsigned bitsEncontrados(unsigned word[N],char bit){
int b=0;
int i;
for (i=0;i<N;i++)
        if(word[i]==0 && bit=='0'|| word[i]==1 && bit=='1') b++;
return b;
}
```

IDENTIFICADOR ESTUDIANTE

TALLER DE PROGRAMACIÓN

EXAMEN DEL PRIMER PARCIAL

22 OCTUBRE 2021

ASIENTO

APELLIDOS (en MAYÚSCULAS)

NOMBRE (en MAYÚSCULAS)

APARTADO 2 (1 puntos)

Dado el código siguiente, completar debajo de cada comentario la instrucción que falta.

```
int main(){
   unsigned w[N]; char b;

// iniciar el algoritmo de generación de números aleatorios. DEBAJO DE ESTE COMENTARIO HAY QUE PONER LAS INSTRUCCIONES CORRESPONDIENTES
srand(time(NULL));

// invocar a rndWord. DEBAJO DE ESTE COMENTARIO HAY QUE PONER LAS INSTRUCCIONES CORRESPONDIENTES

randWord(w);

// invocar a scanBit. DEBAJO DE ESTE COMENTARIO HAY QUE PONER LAS INSTRUCCIONES CORRESPONDIENTES
scanBit (&b);

// printar el valor obtenido al invocar a bitsEncontrados. DEBAJO DE ESTE COMENTARIO HAY QUE PONER LAS INSTRUCCIONES CORRESPONDIENTES
printf("%i",bitsEncontrados(w,b));

return 0;
}
```