

TALLER DE PROGRAMACIÓN

EXAMEN

20 ENERO 2022

PARCIAL II, PROBLEMA 1 (2 puntos)

Se da la definición de una lista simplemente enlazada y unos prototipos.

```
struct Tnodo {
    int dato;
    struct Tnodo * siguiente;
};

struct TlistaNodos {
    struct Tnodo * inicio;
};

void iniciarLista(struct TlistaNodos *);

void insertarListaCircular (int, struct TlistaNodos *);
// crea un nodo, cuyo campo dato es un integer e inserta dicho nodo en una lista
```

Se debe implementar la función de prototipo **void recorrerListaCircular(struct TlistaNodos)** que recorre una lista y printa el contenido del campo dato de cada nodo, y aplicar dicha función en el main, utilizando los prototipos dados.

```
#include "stdio.h"

struct Tnodo
{
    int dato;
    struct Tnodo * siguiente;
};

struct TlistaNodos {
    struct Tnodo * inicio;
};

void recorrerListaCircular (struct TlistaNodos listaNodos);

int main (){
    struct TlistaNodos listaNodos;
    iniciarLista (&listaNodos);
    insertarListaCircular(1,&listaNodos);
    recorrerListaCircular(listaNodos);
    return 0;
}

void recorrerListaCircular (struct TlistaNodos listaNodos)
{
    struct Tnodo * nodo;
    nodo=listaNodos.inicio;
    if (nodo!=NULL)
        do{ printf("%d ",(*nodo).dato ); // nodo->dato
            nodo =(*nodo).siguiente; // nodo->siguiente
        } while(nodo!=listaNodos.inicio);
}
```

PARCIAL II, PROBLEMA 2 (2 puntos)

Se da un código a continuación, con la definición de la primera parte de una apuesta de la Primitiva, donde los dos primeros caracteres son “28”, los siguientes seis son dígitos y el siguiente una letra, que se han de generar aleatoriamente, sin usar ninguna otra función definida por el programador.

```
typedef struct
{ char identificador[9+1];
  // unsigned numeros[N];
} APUESTA;
```

Se debe implementar la función con prototipo **APUESTA apuesta_generada()** y printar, en el main, el identificador generado.

```
#include "stdio.h"
#include "stdlib.h"
#include "time.h"
```

```
typedef struct
{ char identificador[9+1];
  // unsigned numeros[N];
} APUESTA;
```

```
APUESTA apuesta_generada();
```

```
int main()
{ srand(time(NULL));
  printf("%s",apuesta_generada().identificador);
  return 0;
}
```

```
APUESTA apuesta_generada()
{
    APUESTA apuesta;
    apuesta.identificador[0] = '2';
    apuesta.identificador[1] = '8';
    int j;
    for (j=2; j<8; j++)
        apuesta.identificador[j] = rand()%10+'0';
    apuesta.identificador[j++] = rand()%26+'A';
    apuesta.identificador[j] = '\0';
    return apuesta;
}
```

TALLER DE PROGRAMACIÓN

EXAMEN

20 ENERO 2022

PARCIAL II, PROBLEMA 3 (2 puntos)

Se da la definición de la primera parte de una apuesta de la Primitiva, donde los dos primeros caracteres son "28", los siguientes seis son dígitos y el siguiente una letra, generados aleatoriamente.

```
typedef struct
{ char identificador[9+1];
  // unsigned numeros[N];
} APUESTA;
```

Se debe implementar el main que leyendo el fichero "apuestas.txt", que se ha generado aparte, copie su contenido en el fichero binario "apuestas.dat". Una vez terminada esta primera tarea, se deberá leer el fichero binario, copiando de nuevo su contenido en el fichero de texto, pero eliminando los dos primeros caracteres ("28").

```
#include "stdio.h"
#include "stdlib.h"
```

```
typedef struct
{ char identificador[9+1];
  // unsigned numeros[N];
} APUESTA;
```

int main()

```
{ APUESTA apuesta;
  FILE * f_in, * f_out;
  f_in=fopen("apuestas.txt", "r");
  if(f_in==NULL) {
    printf("no se pudo abrir apuestas.txt");
    exit(1); }
  f_out=fopen("apuestas.dat", "wb");
  while (fscanf(f_in,"%s",apuesta.identificador)!=EOF)
    fwrite(&apuesta,sizeof(APUESTA),1,f_out);
  fclose(f_in);
  fclose(f_out);

  f_in=fopen("apuestas.dat", "rb");
  if(f_in==NULL) {
    printf("no se pudo abrir apuestas.txt");
    exit(2); }
  f_out=fopen("apuestas.txt", "w");
  while (fread(&apuesta,sizeof(APUESTA),1,f_in)>0)
    fprintf(f_out,"%s\n",apuesta.identificador+2);
  fclose(f_in);
  fclose(f_out);
  return 0;
}
```

PARCIAL I, PROBLEMA 1 (2 puntos) EXCLUSIVAMENTE EN EVALUACIÓN POR “SÓLO PRUEBA FINAL”

Se debe implementar la función que tiene como prototipo **void rotarBitsDchaCircular(char bits[N])** que rota circularmente hacia la derecha los bits de un número binario de longitud N, e invocar a dicha función desde main, donde tenemos la declaración local *char bits[N]*, cuyo valor es indiferente.

```
#define N 4

void rotarBitsDchaCircular(char bits[N]) ;

int main()
{ char bits[N];
  rotarBitsDchaCircular(bits);
  return 0;
}

void rotarBitsDchaCircular(char bits[N])
{ char bitDcha= bits[N-1];
  for (int i=N-1;i>=1;i--)
    bits[i]=bits[i-1];
  bits[0]=bitDcha;
}
```

PARCIAL I, PROBLEMA 2 (2 puntos) EXCLUSIVAMENTE EN EVALUACIÓN POR “SÓLO PRUEBA FINAL”

Se debe implementar la función que tiene como prototipo **void scanBits(char bits[N])** que lee desde teclado los bits de un número binario de longitud N, y la función de prototipo **void randBits(char bits[N])** que genera aleatoriamente dicho número binario.

```
void scanBits(char bits[N]){
  for (int i=0;i<=N-1;i++){
    do {
      fflush(stdin);
      printf("bits[%i]? ",i);
      scanf("%c",&bits[i]);
    } while (!(bits[i]=='0' | bits[i]=='1'));
  }
}

void randBits(char bits[N])
{ for (int i=0;i<=N-1;i++)
  bits[i]=rand()%2+'0';
}
```