

Programación de Computadores

Resolución Prueba N° 1

(Capítulos 1, 2 y 3 : Lenguaje C y Estructuras de Control)

Recuerde que muchas formas de resolver un problema, los ejercicios en esta pauta son para usar de referencia, los problemas aquí resueltos no son las alternativas más eficientes, pero si son bastante entendibles y fáciles de ver.

1.- Mano de Truco:

(2.2 ptos.)

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

int main()
{
    int carta1, pinta1, carta2, pinta2, carta3, pinta3;
    srand(time(0));
    carta1=rand()%10+1;
    pinta1=rand()%4+1;
    carta2=rand()%10+1;
    pinta2=rand()%4+1;
    while (carta1==carta2 && pinta1==pinta2)
    {
        carta2=rand()%10+1;
        pinta2=rand()%4+1;
    }
    /* ya tengo dos cartas distintas*/

    carta3=rand()%10+1;
    pinta3=rand()%4+1;
    while ((carta1==carta3 && pinta1==pinta3) ||
           (carta2==carta3 && pinta2==pinta3))
    {
        carta3=rand()%10+1;
        pinta3=rand()%4+1;
    }
    /*aqui ya tengo las tres distintas*/

    printf("Su juego es:\n\n");
    if (carta1>=2 && carta1<=7)
        printf("%d de ",carta1);
    else
        switch(carta1)
        {
            case 1: printf("As de ");
                     break;
            case 8: printf("Sota de ");
                     break;
            case 9: printf("Caballo de ");
                     break;
        }
}
```

```

        case 10: printf("Rey de ");
                break;
    }
    switch(pinta1)
    {
        case 1: printf("Espadas\n");
                break;
        case 2: printf("Oros\n");
                break;
        case 3: printf("Copas\n");
                break;
        case 4: printf("Bastos\n");
                break;
    }
    if (carta2>=2 && carta2<=7)
        printf("%d de ",carta2);
    else
        switch(carta2)
        {
            case 1: printf("As de ");
                    break;
            case 8: printf("Sota de ");
                    break;
            case 9: printf("Caballo de ");
                    break;
            case 10: printf("Rey de ");
                    break;
        }
    switch(pinta2)
    {
        case 1: printf("Espadas\n");
                break;
        case 2: printf("Oros\n");
                break;
        case 3: printf("Copas\n");
                break;
        case 4: printf("Bastos\n");
                break;
    }
    if (carta3>=2 && carta3<=7)
        printf("%d de ",carta3);
    else
        switch(carta3)
        {
            case 1: printf("As de ");
                    break;
            case 8: printf("Sota de ");
                    break;
            case 9: printf("Caballo de ");
                    break;
            case 10: printf("Rey de ");
                    break;
        }
    switch(pinta3)
    {
        case 1: printf("Espadas\n");
                break;
        case 2: printf("Oros\n");
                break;
        case 3: printf("Copas\n");
                break;
        case 4: printf("Bastos\n");
                break;
    }
    return 0;
}

```

Obviamente el programa habría sido mucho más eficiente usando arreglos (próximos capítulos). Los switch podrían haberse cambiado por if. También habría sido mejor utilizar do while en vez de los while, nos habríamos ahorrado algunas lines de código.

Como se puede ver, el problema era de baja complejidad, un poco largo, pero nada más ☺.

La función **srand()** y **rand()** se encuentran en el archivo de cabecera **stdlib.h**, y la función **time()** en **time.h**.

2.- los *inicios y cierre de llaves* están correctamente escritos (terminados).

(2.0 ptos.)

```
# include <stdio.h>

int main()
{
    int cuentaIni=0, cuentaFin=0;
    int car='A';
    int flagError=0;

    while(car!=EOF)
    {
        car=getchar();
        if (cuentaIni<cuentaFin)
        {
            printf("Error!!\n");
            flagError=1;
            break;
        }
        if (car=='{')
            cuentaIni++;
        if (car=='}')
            cuentaFin++;
    }
    if (flagError==1)
        printf("Error en llaves\n");
    else
        if (cuentaIni!=cuentaFin)
            printf("Error en llaves\n");
        else
            printf("No hay error en las llaves\n");
    return 0;
}
```

En esta solución, si en algún momento los **}** superan a los **{**, entonces ya se sabe que hay un error y no es necesario continuar.

(Nota: este ejercicio entro en el examen 2002)

3.- *Suponga por un momento que el lenguaje C no trae el operador módulo. Entonces, escriba un programa que permita ingresar dos números enteros **a** y **b** y que calcule el valor de **a** módulo **b**.*

Utilice el siguiente algoritmo para calcular el módulo:

*Se resta: **a - b**,*

*El resultado se vuelve a restar por **b**, y así sucesivamente hasta que el resultado sea menor que **b**, en ese caso, el modulo es el último resultado.*

(0.8 ptos.)

Solución:

```
# include <stdio.h>

int main()
{
    int a,b,resto;
    printf("Programa que calcula a mod b\n");
    printf("Ingrese a: ");
    scanf("%d",&a);
    printf("Ingrese b: ");
    scanf("%d",&b);

    resto=a;
    while (resto>=b)
        resto=resto-b;

    printf("a mod b es : %d\n", resto);

    return 0;
}
```

(Sin comentarios ☺)

4.- *Escriba un programa en C que lea una serie de números y arroje la cantidad de números leídos, el mayor y el menor, el programa debe terminar cuando el dato ingresado sea primo.*

(1.0 ptos.)

Solución:

Muy tradicional para dar la pauta ☺.