

ASIGNATURA:

ALGORITMO Y ESTRUCTURA DE DATOS

EJEMPLOS DE ESTRUCTURAS EN C

2012

ESTRUCTURAS EN C

1. Crear un programa que contenga los siguientes elementos:

Una estructura llamada Estudiante que represente a un estudiante, que tiene como datos: nombre, sexo, edad y un arreglo unidimensional de 5 elementos, correspondientes sus notas. El programa deberá calcular e imprimir cuál es el estudiante con mayor promedio de notas.

```
#include <stdio.h>
#include<string.h>
#include<conio.h>
#include<stdlib.h>
#define NN 5
typedef struct
{
    char *nomest;          // nombre estudiante
    int edad;
    char sexo;
    int *notas;
} Estudiante;

typedef struct
{
    int numest;
    Estudiante *estu;
}RegistroNotas;

void main ()
{
    char Nombre[30];
```

```
RegistroNotas rest;
int i,n,sumnot=0,indicemay=0;
float prom,mayorp=0;

printf ("Cuantos Estudiantes?\n");
scanf ("%d", &rest.numest);
rest.estu=(Estudiante *) malloc(rest.numest * sizeof(Estudiante));
if(rest.estu==NULL)
{
    printf("Insuficiente Espacio de Memoria");
    exit(1);
}

for (i=0; i<rest.numest; i++)
{
    printf ("Nombre Estudiante[%d]: ",i+1);
    fflush(stdin);
    gets(Nombre);

    rest.estu[i].nomest=(char *)malloc(sizeof (char) * (strlen(Nombre)+1));
    strcpy (rest.estu[i].nomest, Nombre);
    printf ("Edad[%d]: ",i+1);
    fflush(stdin);
    scanf ("%d",&rest.estu[i].edad);
    printf ("Sexo[%d]: ",i+1);
    fflush(stdin);
    scanf ("%c",&rest.estu[i].sexo);
    rest.estu[i].notas=(int *)malloc(NN *sizeof(int));
    if(rest.estu[i].notas==NULL)
    {
        printf("Insuficiente Espacio de Memoria");
    }
}
```

```
        exit(1);
    }
    printf("Las 5 calificaciones del estudiante %d son\n",i+1);
    sumnot=0;
    prom=0;
    for(n=0;n<5;n++)
    {
        printf("Nota [%d]: ",n+1);
        scanf("%d",&rest.estu[i].notas[n]);
        sumnot+=rest.estu[i].notas[n];
    }
    prom=sumnot/5;
    if(prom>mayorp)
    {
        indicemay=i;
        mayorp=prom;
    }
}
printf("El estudiante con mayor promedio es:%s\n",rest.estu[indicemay].nomest);
free(nomest);
free(estu);
free(notas);
}
```

2. Un zoológico desea guardar los datos de los pandas. Para ello crear una estructura con lo siguiente: nombre, peso, año de nacimiento, sexo. Cree una función para leer los datos de la estructura.

```
void LeerPanda(Panda *p);
```

Determinar la edad del panda y si puede tener bebés, sabiendo que para poder tener bebés debe ser mayor que 5 años y (obvio) ser hembra. Utilizar las siguientes funciones:

```
int EdadPanda(Panda p); devuelve la edad
```

```
int TenerBebes(Panda p); //devuelve 1 si puede tener bebes y 0 si no
```

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

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

```
#include<string.h>
```

```
typedef struct Panda
```

```
{
```

```
    char *nombrep;
```

```
    float peso;
```

```
    int anyonac;
```

```
    char sexo;
```

```
}mip;
```

```
void LeerPanda(mip *p);
```

```
int EdadPanda(mip p); //devuelve la edad
```

```
int TenerBebes(mip p); //devuelve 1 si puede tener bebes y 0 si no.
```

```
void main()
```

```
{
```

```
    mip mp;
```

```
    int puede;
```

```
    LeerPanda(&mp);
```

```
    puede=TenerBebes(mp);
```

```
    if(puede==1)
        printf("El panda %s puede tener bebes\n",mp.nombrep);
    else
        printf("El panda %s no puede tener bebes\n",mp.nombrep);
}
```

```
void LeerPanda(mip *p)
{
    char nom[35];

    printf("El panda se llama: ");
    gets(nom);
    p->nombrep=(char *)malloc(sizeof(char) * strlen(nom)+1);
    if(p==NULL)
    {
        printf("Insuficiente espacio de memoria\n");
        exit(1);
    }
    strcpy(p->nombrep,nom);
    fflush(stdin);
    printf("Su peso es:");
    scanf("%f",&p->peso);
    fflush(stdin);
    printf("Su año de nacimiento es:");
    scanf("%d",&p->anyonac);
    fflush(stdin);
    printf("Su sexo es:");
    scanf("%c",&p->sexo);
}
```

```
int EdadPanda(mip p)//devuelve la edad
```

```
{  
    int edadp;  
    edadp=2012 - p.anyonac;  
    return(edadp);  
}
```

```
int TenerBebes(mip p) //devuelve 1 si puede tener bebes y 0 si no.
```

```
{  
    int res,edad;  
    edad=EdadPanda(p);  
    if(edad>=5 && (p.sexo=='f' || p.sexo=='F'))  
        res=1;  
    else  
        res=0;  
    return(res);  
}
```

3. Complete los espacios en blanco en el main y escriba el código de la función LeerFraccion y SumarFracciones. Este programa deberá sumar dos fracciones y mostrar el resultado de dicha operación por pantalla.

```
#include <stdio.h>
#include <stdlib.h>
typedef struct racional
{
    int numerador;
    int denominador;
}fraccion;

void LeerFraccion(fraccion *pf);           //prototipo de la función
fraccion *SumarFracciones(fraccion f1, fraccion f2); // prototipo de la función

void main(void)
{
    fraccion fr1, fr2, *fr3;
    LeerFraccion(&fr1); //Llamar a la función LeerFraccion
    LeerFraccion(&fr2); //Llamar a la función LeerFraccion
    fr3=SumarFracciones(fr1,fr2); //Llamar a la función SumarFracciones
    printf("\n\n%d/%d      +      %d/%d      =      %d/%d
\n",fr1.numerador,fr1.denominador,fr2.numerador,fr2.denominador,      fr3-
>numerador, fr3->denominador);
    free(fr3);
}
```

//Definición de la función para leer una fracción

```
void LeerFraccion(fraccion *pf)
{
    printf("Numerador:");
    scanf("%d",&pf->numerador);
    printf("Numerador:");
    scanf("%d",&pf->denominador);
}
```

//Definición de la función para sumar dos fracciones

```
fraccion *SumarFracciones(fraccion f1, fraccion f2)
```

```
{
    fraccion *pf;
    pf=(fraccion *) malloc(sizeof(fraccion));
    if( pf == NULL )
    {
        printf("Memoria insuficiente\n");
        exit(-1);
    }
}
```

```
pf-
```

```
>numerador=(f1.numerador*f2.denominador)+(f1.denominador*f2.numerador);
```

```
pf->denominador=f1.denominador*f2.denominador;
```

```
return (pf);
```

```
}
```