

PRÁCTICA 3: ESTRUCTURAS Y PUNTEROS

Cátedra Programación II

Octubre 2023

1. Estructuras

EJERCICIO 1. Considere la siguiente estructura que representa un punto en el plano.

```
1 struct punto {  
2     double x;  
3     double y;  
4 }
```

Defina las siguientes funciones:

1. Función proyección:

```
1 // Recibe un punto y devuelve la proyección sobre el eje  
2 // que se pida ('x' o 'y'). Cualquier otro valor debe ser  
3 // invalidado  
4 double proy(struct punto p, char eje){  
5     ...  
6 }
```

2. Función distancia:

```
1 // Devuelve la distancia entre los puntos recibidos  
2 double dist(struct punto punto1, struct punto punto2){  
3     ...  
4 }
```

3. Función pos:

```
1 // Devuelve '1', '2', '3' o '4' si el punto se encuentra  
2 // estrictamente en un cuadrante, 'x' o 'y' si se encuentra  
3 // en uno de los ejes u 'o' si es el origen de coordenadas  
4 char pos(struct punto p){  
5     ...  
6 }  
7
```

2. Punteros

EJERCICIO 2. Dadas las siguientes declaraciones:

```

1 int vector[5] = {10, 20, 30, 40, 50};
2 int a = 3;
3 int *ptr = &a;
4 int *qtr = vector;

```

Determinar el tipo y valor de:

| | | | |
|-------|--------|-----------|---------|
| a | &a | *a | ptr |
| &ptr | *ptr | qtr | &qtr |
| *qtr | vector | &vector | *vector |
| ++qtr | ++*qtr | ++*vector | *&ptr |

EJERCICIO 3. Dadas las siguientes declaraciones:

```

1 int *ip1, ip2;
2 char ch, *cp;

```

¿Cuáles de las siguientes asignaciones son violaciones de tipo? Explique el motivo.

| | | |
|----------------------------|------------|------------|
| ip1 = "cadena de ejemplo"; | cp = 0; | ip1 = 0; |
| cp = &'a'; | ip1 = ip2; | cp = '\0'; |
| ip1 = '\0'; | cp = &ch; | *ip = ip2; |

3. Arreglando programas

EJERCICIO 4. Cada uno de los siguientes fragmentos de programas tienen, al menos, un error y/o alguna construcción no recomendable. Identifíquelos y de alternativas para su re-escritura.

1. Alcance (*scope*) de variables:

```

1 #include <stdio.h>
2 int a, b;
3 void initialize() {
4     if (a>0) b=1; else b=0;
5 }
6 int main() {
7     a=1;
8     initialize();
9     printf("%d %d\n", a, b);
10    return 0;
11 }

```

2. Punteros:

```
1 #include <stdio.h>
2 int main() {
3     int *pi = malloc(sizeof(int));
4     int *pj = malloc(sizeof(int));
5     *pi = 11;
6     pj = pi;
7     printf("*pi=%d, *pj=%d", *pi, *pj);
8     free(pj);
9     // resto de código
10    return 0;
11 }
```

3. Punteros:

```
1 #include <stdio.h>
2 int main() {
3     int *ptr, a=34;
4     int *ptr2=&b, b=45;
5     *ptr = 34;
6     printf("%d == %d", a, *ptr);
7     return 0;
8 }
```

4. Estructuras:

```
1 #include <stdio.h>
2 struct vector {
3     float x, y;
4 }
5 struct vector *normal(struct vector v) {
6     struct vector *ptr;
7     ptr->x = v.x;
8     ptr->y = v.y;
9     return ptr;
10 }
```

4. Estructuras y Punteros

EJERCICIO 5. Dadas las siguientes definiciones de estructura y asignaciones:

```
1 struct estudiante {  
2     char *apellido;  
3     int anio_ingreso;  
4     char division;  
5 };  
6 struct estudiante nuevo, *pest = &nuevo;  
7 nuevo.apellido = "Lopez";  
8 nuevo.anio_ingreso = 1998;  
9 nuevo.division = 'A';
```

Determine si las siguientes sentencias son correctas. En caso de ser incorrectas explique por qué, en caso de ser correctas indique cuál es el valor que representan.

- a. nuevo->apellido
- b. pest->division
- c. (*pest)->apellido
- d. *pest->apellido+2
- e. *(pest->apellido+2)
- f. pest->apellido[2]