

Guía examen 2 "Programación Estructurada" (oid)

Funciones

(&, *)

char[10], int)
C float*, int[10]
C int[6][6]
C char[5][15]

- * Mandas valores por valor o por referencia
- * Puedes pedir valores en la función o los puedes recibir
- ** Ve si el arreglo lo mandas completo o solo una parte (num) (num[13] [13][10])
- * Lo que regresas tiene que ser del mismo tipo que la función y la variable que lo guardará
- * Donde declaras la función se pone el tipo, al llamarla no \rightarrow de lo que manda/recibe `<time.h>`
- * Si después lo necesito, entonces si lo guardo `<string.h>`
`<ctype.h>`
- * fflush antes de char a leer \rightarrow tam \rightarrow última posición
- * Puedes moverte solo en el que pongas `pal[5][15]` \rightarrow gets(`pal[1]`)
- * If C {return r;} else {return -1;} para validar en main \rightarrow corto
- * Si lleno con caracteres y quiero imprimir o usar como cadena, necesita '\0'
- * Para inicializar `v[10] = {0}` `verbo[10] = {"comer"}`

Aprovecha

Ciclos

¿Cómo

simplifico

el

programa?

↓

main

corto

- * No puedo comparar caracteres con cadenas
- * En cadenas: Ver que copie, que concateno y que yo tenga predefinido
- * Llegar a una dirección con lo que pido/quiero `a[0][2][0]` `[azor]` `[grupo]`
- * En ciclos piensa cómo se repite y cómo se sale
- * Char puede contener letras o números

Aprovecha

arreglos

- * Recursividad: Acercamiento y condición de corte \rightarrow llamada recursiva estado base \rightarrow num + suma(num-1)

Estructuras

tipo de dato complejo

void llenar(struct persona)
llenar(a)

struct

strcpy

strcmp

strlen

- * Una variable o un arreglo pueden ser de tipo struct dependiendo de cuantas estructuras necesites o cuantas veces necesites entrar
- * Puedes mandar el arreglo de estructuras y llenar o puedes crear una variable local y regresar la estructura al arreglo en el main
- * Si quiero modificar solo un campo: `ver[13].edad` `int duplicacion (struct persona per)`
`per[13].edad = duplicacion(per[13])` \rightarrow return `per.edad;`
- * Verifica a que estructura y campos te refieres, `Cstruct persona vec[10]`
pon la dirección específica: ¿Estructura? ¿Solo campo?
- * Conserva cuantas veces has entrado a la estructura \rightarrow tamaño del vector

- General a particular ¿quién tiene a quién? ¿Qué está adentro? → caminito
- Checka puntos**
- * ① Qué datos necesito ② Qué tengo que devolver *type of struct, param;*
 - * Si solo es una cosa, no hay que guardarla en una estructura
 - * Valida que no sobrescriba, que no rompa ninguna regla, condicionalo
- for $Ci=0; i < \text{dir}; i++$ → hasta donde he llenado, llenar con referencia
- %s → .nombre %d → .edad • Estructuras anidadas

Inicializa variables **Instrucciones sencillas** **Automatizador**

con o sin struct

- * Uso de banderas para validar



*void ordenar(Cstruct profesor prof[30])
ordenar(Csector[20], profesores)*

Apuntadores

num, v[20], m[20]
 $ap = \& \text{donde apunta}$ * $ap \rightarrow \text{lo que tiene la dirección}$

- * En arreglos si puedes avanzar la dirección a la que apunta *dirección (%p, ap)*
- if (regap > repetidos[20]) → si apuntan a lo mismo o no
- * Desde la declaración es un apuntador y tiene tipo de dato
- * Puedes mandarlo a una función sin forzarlo *funcion(ap) void funcion(int *ap)*
- * $Cap + 1 = \text{rand}() \% 40$ *apf = (Cap + a)
for (; apv <= apv1; apv++)*
- * Recorrer un vector con dos apuntadores

- * Si llenas en una función con un apuntador, ya no necesitas enviar el vector
- int *ap = NULL → inicializar if (Capv < apv2) → checas direcciones
- scanf("%d", &*ap); (*ap)++*

- * STACK Filo // Lifo  elimina último Pila *Push L+1=F
Pop Finis L-1*
- * QUEUE FIFO  Cola, Torroide (primera y ultima consecutivos)
- gets(Cap + *i) → name, app) scanf("%d", &(Cap + *i)) → grupo)*

- * Puedes pasar una estructura completa con *apfin = al tpo
- $ap[3] \rightarrow ap[INI], ap[AUX], ap[Fin]$

Archivos

- * En los archivos de texto (solo texto) (si formato), en los binarios (cualquier tipo) (no formato) (estruct)
- r (lectura) w (escritura) a (añadir) b (binario) + (lectura/escritura) *FILE *ap*
- Lap inicio Lap donde quedo ap = fopen("almamres.bin", "ab");*
- fopen, fclose, fseek, fwrite y fread, fprintf y fscanf, ftell, *fclose(ap)*
- fgetc, fputc, feof, rewind, fflush, remove, ferror
- while (!feof(file)) if(file), if(file == NULL)*

fprintf(file, "%s\n", linea); fscanf(archivo, "%s", &nombre[i]);

fputs(linea, file); fgets(producto[i], 100, archivo);

fwrite(&x, sizeof(struct libro), 1, ap); fread(&x, sizeof(L), 1, ap);