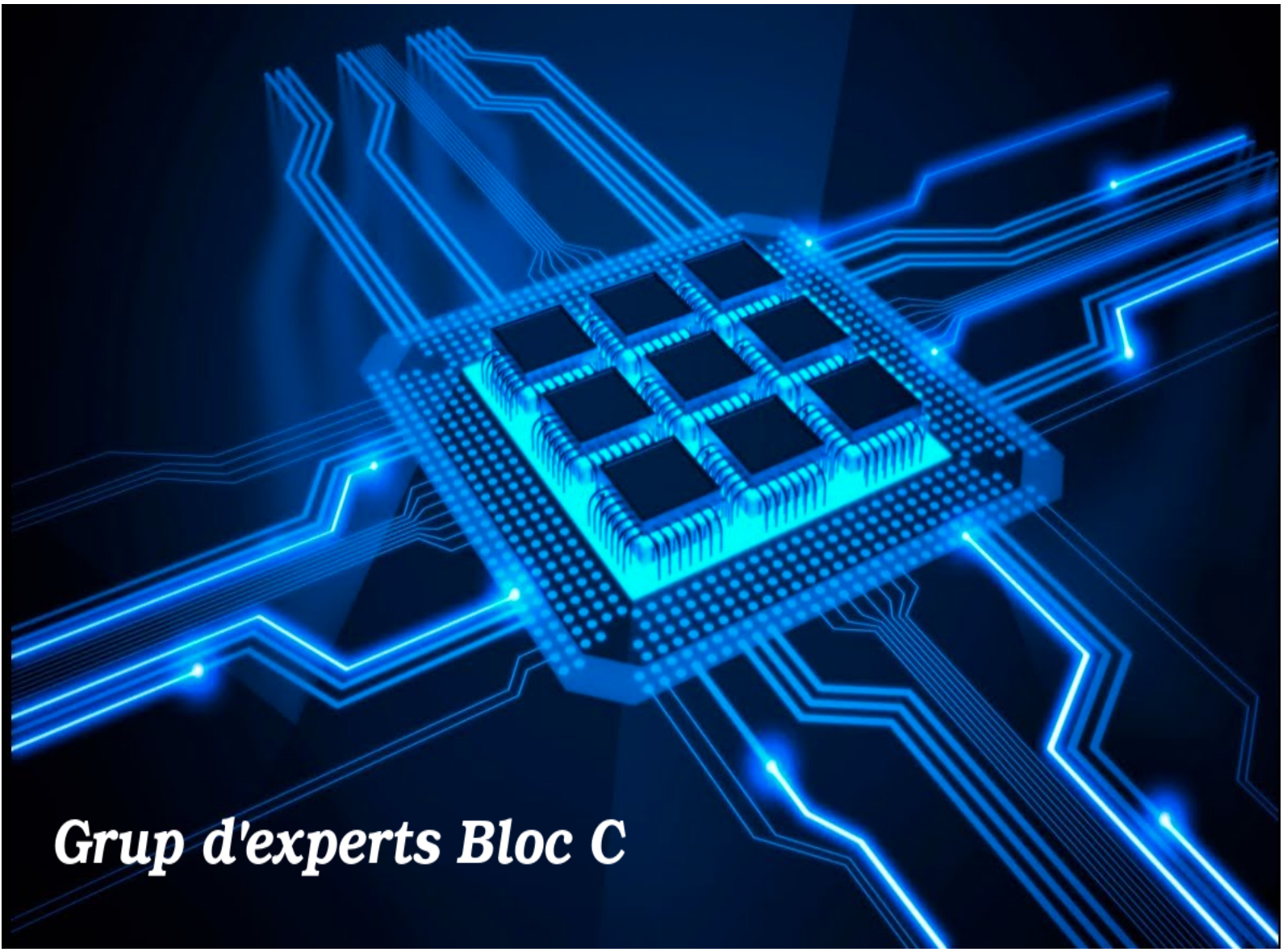

1r ASIX
MP03 - Programació bàsica

Omar El Balaoui
Hugo Martín
Max Segura
Andreu Monforte

Institut de l'Ebre
Tortosa

INDEX

1. Blocs d'un programa informàtic

Existeixen dos tipus de blocs els quals formen un programa informàtic:

Bloc de declaracions: En aquest bloc s'expliquen detalladament tots els objectes que fara servir el programa (arxius, variables, constants…)

Bloc d'instruccions: Aquest bloc es un conjunt de operacion o accions les quals s'han de realitzar per aconseguir els resultats que s'esperen.
Aquest bloc està compost per tres parts, encara que depenent de la situació no sempre estan delimitades de la mateixa manera, aquestes parts son:

- Entrada: La entrada es la part del inici del programa, osigui en el que recull les dades.
- Algoritme: És la part de procés del programa, osigui en el que fa els càlculs.
- Sortida: La sortida es el final del programa, quan s'obtenen els resultats.

Estructura de un programa informático

Cabecera	A modo de comentarios se suele especificar: • Nombre del programa • Datos de entrada • Datos de salida
Funciones	Definición de funciones propias creadas por el programador para usarlas en varias ocasiones
Declaraciones	Definiciones y tipos de: • variables • constantes • nuevos tipos de datos
Asignaciones	Valores iniciales de los identificadores declarados previamente
Entradas	Instrucciones para almacenar en memoria los valores de algunos identificadores
Control	Instrucciones de control de flujo del programa. Pueden ser: • Alternativas • Repetitivas
Salidas	Instrucciones para devolver los resultados obtenidos

A continuació, veurem l'estructura bàsica d'un programa informàtic:

- **Capçalera:** se sol especificar el nom del programa i les dades d'entrada i de sortida.
- **Funcions**: definició de les funcions pròpies crades pel programador per a utilitzar-les en repetides ocasions.
- **Declaracions:** definicions i tipus de variables, constants i nous tipus de dades.
- **Assignacions:** valors inicials dels identificadors declarats prèviament.
- **Entrades:** instruccions per a guardar en la memòria els valors d'alguns identificadors.
- **Control**: instruccions de control de flux del programa que poden ser alternatives o repetitives.
- **Sortides:** instruccions per a retornar els resultats obtinguts.

2.4.2. Tipus de dades compostes

- **Que es una dada composta?**

Entre tots els tipus de dades compostes hi ha molta varietat: registres, vectors, objectes, cadenes, mapes… però la quantitat, els noms, i les definicions variaran depenen el llenguatge i el disseny que s'utilitzi.

- **Tipus de dades compostes:**

- **Predefinits:** El tipus string o també anomenat "Cadena de caràcters" es una secuencia de caracteres que es tractada com una única data.

exemple:

```
typedef char cadena[5]
cadena str1, str2
```

- **Definits:** Hi han dos tipus de dades definits i son els següents:

    - ***Struct:*** També anomenat estructura o registre es una agrupació d'elements de diferents tipus, cada camp es representa mitjançant un caràcter propi.

exemple:

```
struct Nom
{
        Tipus Nom_camp1;
        Tipus Nom_camp2;
}
```

    - ***Array:*** Un array o vector serveix per a agrupar variables d'un mateix tipus amb el mateix nom.

exemple:

```
Tipus Nom [NumElements]
```

2.5. Estructures de selecció, repetició i salt

Estructures de selecció

- **Simples(if):** L'objectiu d'aquesta es triar entre executar un bloc de codi o no.

Aquestes es componen per tres parts:

- Expressió condicional: Pot ser una expressió relacional o boolean, el resultat d'aquesta determina si s'executa un bloc o no.
- Bloc d'accions: Aquest és una o un conjunt d'accions que només s'executaran si l'anterior expressió és correcta.
- Secció comú: Aquest bloc s'executarà sempre que s'execute la expressió condicional donant igual el resultat d'aquesta.

- **Doble(if-else):** Aquesta té l'objectiu de triar entre diferents blocs de codi per a executar.

Té les següents parts:

- Expressió condicional: Aquesta és la que determina quin bloc (Verdader o Fals) executará.
- Bloc d'execució (Verdader): Aquest bloc només s'executara en el cas de que la expressió es verdadera.
- Bloc d'execució (Fals): Aquest bloc només s'executara en el cas de que la expressió es fals.
- Secció comú: Aquest bloc s'executarà sempre que s'execute la expressió condicional donant igual el resultat d'aquesta.

- **Multiple(switch):** Aquesta es permet fer una selecció entre diferents opcions, Es sol utilitzar quan ens trobem amb tantes opcions que no es pot utilitzar una estructura de selecció doble.

Aquestes tenen les següents parts:

- Variable: Aquesta és la base d'on es comparen amb les diferents opcions.
- Opcions: Les diferents opcions que la variable pot triar i que ens tria elegir.
- Bloc d'opcions: Bloc d'opcions de cada una de les opcions.
- Bloc d'accions: Aquest correspon a la opció default.
- Secció comú: Aquest bloc s'executarà sempre que s'execute la expressió condicional donant igual el resultat d'aquesta.

Estructures de repetició

- **WHILE:** Aquesta es en la que el número de repeticions de bucle no es coneixen y es repeteix mentres es compleix una condició.
  Si la condició es verdadera s'executa i torna al principi del bucle, pero si la condició es falsa no s'executa el bloc.

- **DO WHILE:** La diferencia entre aquest i el "WHILE" és que aquest evalua la condició abans d'executar el bloc mentres que do-while fa el contrari - evalua la condició després d'executar el bloc d'instruccions.

- **FOR:** permet repetir un nombre determinat de vegades un conjunt d'instruccions.

Estructures de salt

- **BREAK:** Aquesta permet acabar l'execució d'una estructura condicional, així el nostre programa segueix amb la següent instrucció.

- **CONTINUE:** Continue serveix per parar un bucle a diferencia del break aquest a l'hora de parar el bucle ens portarà de nou al bucle. Aquesta sempre s'utilitza dins d'un bucle.

- **RETURN:** Amb aquesta sentència podem parar qualsevol mètode o funció i així tornar al la instrucció que va crear aquesta funció.

2.6. Tractament de cadenes

- **Que es una cadena?**

Una cadena de caracters o string es una secuencia de caracters ordenats amb longitud indefinida.

- **Operacions amb cadenes.**

- Assignació: Assignar una cadena a una altra.
- Concatenació: Unir una cadena o més per crear una cadena major.
- Cerca: Localitzar dins d'una cadena una subcadena o caràcter.

- Extracció: Extreure d'una cadena qualsevol element ja sigui caracter com un altra cadena.
- Comparar dues o més cadenes.

- **Representació de cadenes.**

La representació de cadenes es comença utilitzant la paraula char (Cadena en inglés) a continuació introduïm el nom de la cadena i per finalitzar posem el caràcter que volem introduir entre comilles simples i si vol una paraula entre comilles dobles.

```
char text[] = 'A';
char text[] = "adeu";
```

- **Tipus de cadenes.**

- Dinàmiques: Es pot alterar la llargada durant el temps d'execució.
- Estàtiques: La llargada es fixa al llarg del temps d'execució.

- **Funcions de tractament de cadenes:**

- **strcpy:** Aquesta funció s'utilitza per copiar funcions d'una ubicació a una altra ocupada per un cadena diferent. Aix vol dir que la funció sobre la que s'ha copiat una cadena es queda destruïda.

strcpy(<variable_destí>,<variable_font>);

- **strcat:** Aquesta a diferencia de "strcpy" al moment de fer una copia sobre una cadena no la destrueix sinó que copia la cadena detrás de la cadena de destí..

strcat(<variable_destí>,<variable_font>);

- **strlen:** Ens permet mostrar el total de caràcters d'una cadena.

<variable>=strlen(<cadena>);

- **strcmp:** La funció de comparació "strcmp" s'utilitza per fer la comparació entre cadenes ja que en el moment de fer una comparació en cas de que la A sigue mes gran que la B ens retorna ens retorna un numero menor a 0, si al B es menor a la A ens retorna 0 i per últim si les dos cadenes son iguals ens retorna un numero major a 0

*int strcmp(const char *s1,const char *s2);*

- **isalum:** Ens retorna "cert" en el cas de que la variable sigue una letra o un digit i fals en el cas contrari:

int isalum (int c);

- **isalpha:** Ens retorna “cert” si el caracter elegit es tracta d’una letra en el cas contrari ens retorna “fals”.

int isalpha (int c);

- **isblank:** Ens retorna “cert” si el caracter es un espai en blanc en cas contrari ens retorna fals.

isblank(<valor>);

- **isdigit:** Ens retorna “cert” si el caracter es un digit en cas contrari ens retorna fals.

int isdigit(int c)

- **isspace:** Ens retorna “cert” si el caracter es un espai en blanc,un salt de linea, tabulador etc.. en cas contrari ens retorna fals.

int isspace(int c);

- **toupper:** Ens retorna la majúscula asociada al carácter en cas de no tenir majuscula ens retorna el caracter complet.

int toupper(int c);

Bibliografia

- http://agora.pucp.edu.pe/inf2170681/2.htm
- http://dis.um.es/~lopezquesada/documentos/IES_1415/IAW/curso/UT3/ActividadesAlumnos/its java7/paginas/pag2.html
- https://aleph.org.mx/cuales-son-los-bloques-de-programacion
- https://education.microsoft.com/es-mx/course/71058557/3
- https://primer.dynamobim.org/es/07_Code-Block/7-1_what-is-a-code-block.html
- http://tic.taboadaleon.es/Unidad1-Programacion/Tema2_Lenguajes/contenido/5_estructura_de_un_programa_informtico.html
- https://sites.google.com/site/elsaverdelavida/unidad-3-control-de-flujo-de-programa/3-3-control-repetitivo
- https://es.slideshare.net/carlospesrivas/instrucciones-de-control-alternativas
- https://ocw.unican.es/pluginfile.php/293/course/section/228/cap4-datos-compuestos.pdf
- https://ikastaroak.birt.eus/edu/argitalpen/backupa/20200331/1920k/es/DAW/DWES/DWES02/es_DAW_DWES02_Contenidos/website_4_tipos_de_datos_compuestos.html
- https://medium.com/lenguajes-y-dialectos-en-programaci%C3%B3n/los-tipos-de-datos-compuestos-la-taxonom%C3%ADa-estructural-b1e7ff31dc38
- https://jj.github.io/1line-py/txt/04.componiendo.html
- http://www.lcc.uma.es/~vicente/docencia/fundprog/teoria/fp_4_estr.pdf
- https://dcodingames.com/estructuras-de-seleccion/
- https://ikastaroak.birt.eus/edu/argitalpen/backupa/20200331/1920k/es/DAMDAW/PROG/PROG03/es_DAMDAW_PROG03_Contenidos/website_3_estructuras_de_seleccin.html
- https://docplayer.es/46447057-Estructuras-de-seleccion.html
- https://fundamentos.fandom.com/es/wiki/ESTRUCTURAS_DE_REPETICION
- https://www.ecured.cu/Cadena_de_caracteres
- https://arquimedes.matem.unam.mx/mati/actividades/info/info_que_es_una_cadena_de_caracteres/index.html
- https://docplayer.es/69006855-Cadenas-de-caracteres-1-definicion-2-funciones-para-manejo-de-cadenas.html
- http://solucioningenieril.com/programacion_en_c/funciones_para_el_manejo_de_cadenas_de_caracteres
- http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/control/lengua_C/cadenas.htm
- https://sites.google.com/site/2avpgccastlillop/manejo-de-cadenas-en-c