LAB 9 – ACCIONS I FUNCIONS

Objectius

- Aprendre a programar procediments amb pas de paràmetres per valor i referència.
- Implementar procediments a partir de l'especificació i de la definició de la seva capçalera.
- Dissenyar la capçalera d'un procediment i realitzar la seva implementació a partir de l'especificació de la tasca que ha de realitzar.
- Utilitzar procediments amb pas de paràmetres per referència i per valor.

Punt de partida

Apunts del tema 3.

Programació de Procediments

Com ja sabeu, els procediments són parts de codi que s'identifiquen i s'agrupen perquè implementen trossos de programa que es repeteixen molt, o bé que poden ser útils per la seva rellevància en altres programes, o que implementen aplicacions molt concretes.

Ja coneixeu les "llibreries" del C, en les quals tenim agrupades un munt de procediments que són molt útils en els programes típics, per fer coses tant habituals com la interacció de l'usuari amb els perifèrics habituals, o càlculs matemàtics no implementats dins del "C".

De la mateixa manera, com a programadors, ens pot interessar desenvolupar procediments que ens facilitin el disseny dels nostres programes, per la reutilització d'aquests en diferents programes, o per la claredat del codi que estem dissenyant.

A la pràctica que fareu avui dissenyareu una sèrie de petits procediments que utilitzareu per fer un programa sencer, i que serviran per organitzar el codi i fer-lo més entenedor. Abans d'incorporar un procediment en el codi d'un altre programa s'ha de provar de manera indepedent, és per això que se us demana explicitament que realitzeu un programa principal per a cada procediment.

A l'hora de fer la implementació en C tots els procediments que dissenyareu han d'estar en un únic fitxer (.c). Als apunts del tema 3 trobareu l'explicació de com organitzar els procediments al programa principal.

Exercicis a realitzar per a cada grup de laboratori:
– El grup L1,L2,L3,L4,L5,L6 l'exercici 1
– El grup L7,L8,L9,L10,L11,L12 l'exercici 2
– El grup LL13,L14,L15,L16,L17 l'exercici 3

Feina prèvia a la sessió de laboratori:

Dissenyeu els programes en pseudocodi i el seu corresponent joc de proves.

Feina a realitzar a la sessió de laboratori:

Realitzeu la implementació en C de tots els programes dissenyats i comproveu amb el joc de proves que els programes es comporten com s'ha especificat a l'enunciat.

Exercici 1

1.1. Dissenyeu un procediment que calculi el factorial del nombre que es passa per paràmetre i retorni el calcul realitzat.

La capçalera del procediment ha de ser la següent:
funció Factorial (num: enter) retorna enter
En C correspon a : int Factorial (int num)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

1.2. Programeu un procediment que llegeixi una llista de valors enters pel teclat i calculi quin és el nombre positiu mes petit.

La capçalera del procediment ha de ser la següent:
acció positiu_petit (var posp: enter)
En C correspon a : void positiu_petit (int *posp)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

1.3. Dissenyeu un procediment que calculi les combinacions de m elements agafats de n en n, els valors m i n es passen per paràmetre. La fórmula per calcular les combinacions és:

$C_m^n = \binom{n}{m} = \frac{m!}{n!(m-n)!}$ en cas de que m≥n. Si m< n el resultat serà 0.

Utilitzeu el procediment dissenyat al primer apartat.
La capçalera del procediment ha de ser la següent:
funció Combinacions (m:enter, n: enter) retorna enter
En C correspon a : int Combinacions (int m, int n)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

1.4. Dissenyeu un procediment que ens permeti mostrar per pantalla un menú d'opcions. El menú que ha de mostrar ha de ser el següent:

1 – Calcular el factorial d'un numero
2 – Calcular positiu mes petit
3 – Calcular les combinacions
4 – Sortir del programa
Quina opció vols?:_

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

1.5. Dissenyeu un programa que permeti realitzar tots els càlculs que heu programat amb els procediments anteriors. El programa ha de treure per pantalla el menú d'opcions i ha de mostrar a la pantalla els resultats de les operacions que l'usuari li demani, fins que l'usuari vulgui sortir del programa per l'opció adequada del menú.

Donat que els jocs de proves per provar cadascuna de les operacions que fa al programa ja les heu dissenyat en els apartats anteriors, ara només caldrà dissenyar la resta de proves necessàries per verificar que el programa realitza l'acció esperada.

Exercici 2

2.1. Dissenyeu un procediment que calculi el factorial del nombre que es passa per paràmetre i retorni el calcul realitzat.

La capçalera del procediment ha de ser la següent:
funció Factorial (num: enter) retorna enter
En C correspon a : int Factorial (int num)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

2.2. Programeu un procediment que llegeixi una llista de valors enters pel teclat i calculi quin és el nombre negatiu més gran.

La capçalera del procediment ha de ser la següent:
acció negatiu_gran (var negg: enter)
En C correspon a : void negatiu_gran (int *negg)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

2.3. Dissenyeu un procediment que calculi variacions (sense repetició) de m elements agafats de n en n. La fórmula per calcular les variacions (sense repetició) és :

$V_m^n = \frac{m!}{(m-n)!}$ en cas de que m≥n. Si m< n el resultat serà 0.

Utilitzeu el procediment dissenyat al primer apartat.
La capçalera del procediment ha de ser la següent:
funció VariacionsSenseRepeticio (m: enter, n: enter) retorna enter
En C correspon a : int VariacionsSenseRepeticio (int m, int n)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

2.4. Dissenyeu un procediment que ens permeti mostrar per pantalla un menú d'opcions. El menú que ha de mostrar ha de ser el següent:

1 – Calcular el factorial d'un numero
2 – Calcular negatiu mes gran
3 – Calcular les variacions sense repeticio
4 – Sortir del programa
Quina opció vols?:_

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

2.5. Dissenyeu un programa que permeti realitzar tots els càlculs que heu programat amb els procediments anteriors. El programa ha de treure per pantalla el menú programat i ha de mostrar a la pantalla els resultats de les operacions que l'usuari li demani, fins que l'usuari vulgui sortir del programa per l'opció adequada del menú. A més a més heu de dissenyar el joc de proves adequat per provar aquest programa.

Donat que els jocs de proves per provar cadascuna de les operacions que fa al programa ja les heu dissenyat en els apartats anteriors, ara només caldrà dissenyar la resta de proves necessàries per verificar que el programa realitza l'acció esperada.

Exercici 3

3.1. Dissenyeu un procediment que calculi el factorial del nombre que es passa per paràmetre i retorni el calcul realitzat.

La capçalera del procediment ha de ser la següent:
funció Factorial (num: enter) retorna enter
En C correspon a : int Factorial (int num)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

3.2. Programeu un procediment que llegeixi una llista de valors reals pel teclat i calculi quants hi ha entre el rang [0..1].

La capçalera del procediment ha de ser la següent:
acció comptar_rang (var negg: real)
En C correspon a : void comptar_rang (float *negg)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

3.3. Dissenyeu un procediment que calculi les permutacions amb repetició de m elements on es repeteixen els elements a,b,c (el primer element a vegades, el segon b vegades i el tercer c vegades).

La fórmula per calcular les permutacions amb repetició és:

$PR_m^{a,b,c} = m!/(a!b!c!)$ Sent m,a,b,c >0.

Utilitzeu el procediment dissenyat al primer apartat.
La capçalera del procediment ha de ser la següent:
funció PermutacionsAmbRepeticio (m: enter,a: enter,b: enter,c: enter) retorna enter
En C correspon a : int CalcularPermutacionsAmbRepeticio (int m, int a, int b, int c)

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

3.4. Dissenyeu un procediment que ens permeti mostrar per pantalla un menú d'opcions. El menú que ha de mostrar ha de ser el següent:

1 – Calcular el factorial d'un numero
2 – Calcular nombre de valors dintre del rang [0,1]
3 – Calcular les permutacions amb repetició
4 – Sortir del programa
Quina opció vols?:_

Per tal de verificar el correcte funcionament del procediment, dissenyeu el programa principal que crida a aquest procediment i el seu joc de proves.

3.5. Dissenyeu un programa que permeti realitzar tots els càlculs que heu programat amb els procediments anteriors. El programa ha de treure per pantalla el menú programat i ha de mostrar a la pantalla els resultats de les operacions que l'usuari li demani, fins que l'usuari vulgui sortir del programa per l'opció adequada del menú. A més a més heu de dissenyar el joc de proves adequat per provar aquest programa.

Donat que els jocs de proves per provar cadascuna de les operacions que fa al programa ja les heu dissenyat en els apartats anteriors, ara només caldrà dissenyar la resta de proves necessàries per verificar que el programa realitza l'acció esperada.