Semestre 1 - Pràctica 3
Gestió integral dels bombers

Índex

1. Introducció 3
2. Funcionament 4
   2.1 Comandes 6
   2.2 Comanda “show” 7
   2.3 Comanda “add” 8
   2.4 Comanda “operation” 10
   2.5 Comanda “show operations” 12
   2.6 Comanda “exit” 14
3. Consideracions 15
4. Dates d’entrega i avaluació 16

1. Introducció

Després dels dos programes anteriors arriba per fi el lliurament del programa final, el qual ens permetrà fer la gestió integral del parc de bombers, emmagatzemant les dades de totes la persones i equips, així com les operacions d'extinció d'incendis que s'han realitzat. A més l'aplicació permetrà poder consultar tota la informació mitjançant una senzilla interfície de comandes.

OBJECTIU

L'objectiu final d'aquesta pràctica és acabar de consolidar l'entorn de programació, aplicant els conceptes vistos al llarg del semestre, tant d'algorísmica bàsica, estructuració i pas de paràmetres i definició/ús de tipus propis. Concretament, realitzant aquesta pràctica, es pretén :

- Consolidar les sentències i algorísmica bàsica
- Consolidar els coneixements sobre l'estructuració d'un programa
- Consolidar l'ús de procediments
- Consolidar l'ús de funcions
- Consolidar l'ús de paràmetres
- Consolidar l'ús de tipus de dades estructurats (arrays, cadenes i registres)
- Consolidar la definició i ús de tipus propis
- Consolidar la compilació manual d'un projecte en C

2. Funcionament

El programa comença mostrant un missatge de benvinguda i demanant que s'introdueixi la informació inicial que es carregarà en el programa. Aquesta informació descriu la configuració inicial dels equips, indicant per a cada equip el seu identificador, el nombre de bombers que el formen i el nom d'aquests bombers.

El format d'entrada és una cadena configurada de la següent manera:

```
<equip 1>/<equip 2>/<equip 3>/..../<equip n>.
```

Com es pot observar la informació de cada equip se separa pel caràcter '/' i al final de l'últim equip es trobarà el caràcter '.'. Hi haurà un màxim de 8 equips en aquesta cadena i durant l'execució del programa no es podran afegir nous equips.

Cadascun d'aquests equips està format per la següent cadena:

```
<id equip x>/<nombre de bombers equip x>/<nom bomber 1>/<nom bomber 2>...
```

L'identificador de l'equip serà una lletra majúscula, mentre que el nombre de bombers de l'equip serà un dígit comprès entre 1 i 9.

En l'Output 1 podeu veure un exemple vàlid d'introducció de dades.

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City >
```

Output 1. Exemple d'introducció de dades inicials correctes. En aquest exemple es carreguen 3 equips (F, B i D) amb 2, 1 i 1 bombers respectivament.

En cas que l'identificador d'un equip no sigui una lletra majúscula o bé el nombre de bombers no sigui un número comprès entre 1 i 9, s'haurà de donar l'error corresponent i finalitzar l'execució del programa amb un missatge de comiat (veure Output 2). En cas que la cadena contingui diversos errors s'indicaran únicament els del primer equip afectat (pot ser que tingui 1 o 2 errors) i s'ignoraran la resta d'errors de la resta d'equips de la cadena (veure Output 3 i Output 4).

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/0/jonny cash.
(ERROR) Wrong firefighters number

See you later!
```

Output 2. Exemple d'introducció de dades inicials incorrectes. Com el valor 0 no és una dada vàlida, es mostra l'error i es finalitza l'execució del programa.

```
Welcome to Prog City!

Initial info: g/2/jack willis/anne home/B/*/peter jack hero.
(ERROR) Wrong team id

See you later!
```

Output 3. Exemple d'introducció de dades incorrectes. Malgrat que existeix una dada incorrecta en el primer equip i un altre en el segon equip, només es mostren els errors del primer equip i finalitza el programa.

```
Welcome to Prog City!

Initial info: F/1/john cash/h/*/martha king/P/2/mark twice/jenny lynch.
(ERROR) Wrong team id
(ERROR) Wrong firefighters number

See you later!
```

Output 4. Exemple d'introducció de dades incorrectes. En aquest exemple els primers errors es troben en el segon equip, en el qual ocorren els dos errors possibles.

Per a carregar tota aquesta informació es recomana disposar d'un array de registres de tipus Team, dissenyat de la següent manera (es pot observar que existeixen alguns camps relacionats amb les operacions, que s'utilitzaran posteriorment; el camp `n_operations` s'ha d'inicialitzar a 0):

```
#define MAX_NAME               30
#define MAX_FIREFIGHTERS        9
#define MAX_OPERATIONS        100

typedef struct {
   char name[MAX_NAME];
   int n_operations;
   int minutes_op[MAX_OPERATIONS];
} FireFighter;

typedef struct {
   char team_id;
   int n_firefighters;
   FireFighter firefighters[MAX_FIREFIGHTERS];
} Team;
```

Com es pot visualitzar en l'Output 1, una vegada realitzada la càrrega de manera correcta apareixerà l'etiqueta “Prog City >” i el sistema quedarà a l'espera de rebre una comanda per part de l'usuari.

2.1 Comandes

Per a poder utilitzar les diferents funcionalitats desenvolupades en la pràctica, l'usuari podrà executar una sèrie de comandes de text. Així, a diferència de les pràctiques anteriors, en aquest cas no hi haurà cap menú i tota la interacció es realitzarà mitjançant ordres que acceptarà el nostre programa.

Si l'ordre que es rep no figura entre cap de les acceptades pel programa, senzillament s'ignorarà i tornarà a aparèixer l'indicador d'espera de comanda (veure Output 5).

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > hello

Prog City >
```

Output 5. La comanda "hello" no és un de les comandes acceptades pel programa.

2.2 Comanda “show”

La primera comanda i també la més bàsica que es pot executar en el programa és la comanda “show”. Aquesta comanda permetrà saber la composició dels equips i mostrarà per pantalla el nom de l'equip i el nom de les persones que el formen (veure Output 6). L'espai davant de cada membre de l'equip equival a un tabulador (caràcter ‘\t’).

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > show
Team F
        jack willis
        anne james
Team B
        lucas tee warlow
Team D
        johnny crash

Prog City >
```

Output 6. La comanda “show” mostra la composició de cadascun dels equips existents.

Com veurem més endavant hi ha la possibilitat d'utilitzar aquesta mateixa comanda amb alguna variació per a poder mostrar un altre tipus d'informació complementària.

2.3 Comanda “add”

La comanda “add” permet afegir bombers als diferents equips. El format d'aquesta comanda és el següent:

```
add <id equip> <nom del bomber>
```

L'identificador de l'equip correspon al caràcter en majúscula corresponent. Es pot suposar que la dada de l'identificador d'equip que ens introduiran sempre serà un caràcter però, si l'equip indicat no existeix, s'ha de mostrar l'error corresponent (veure Output 7).

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > add Y peter marshall
(ERROR) Team Y does not exist

Prog City > add * mike thompson lee
(ERROR) Team * does not exist

Prog City >
```

Output 7. Exemples d’error a l’introduir un nou bomber en un equip inexistent.

En cas que l'equip sigui correcte, s'afegirà el bomber a l'equip sempre que l'equip no estigui complet (ja existeixen MAX_FIREFIGHTERS per a aquest equip). En cas que l'equip estigui complet es mostrarà també el missatge corresponent (veure Output 8).

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > add F maria

Prog City > add F john rambo

Prog City > add F martha think

Prog City > add F jerry lee lewis

Prog City > add F john john

Prog City > add F mike tyres

Prog City > add F sylvia geek tire

Prog City > add F joe cooker
(ERROR) Team F is full

Prog City >
```

Output 8. L’equip F arriba al màxim de la seva capacitat i es mostra l’error corresponent.

Una vegada afegit el bomber de manera correcta, el programa queda a l'espera de la següent comanda. Es pot observar un exemple complet en l'Output 9.

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > add F jerry lee lewis

Prog City > add B jack willow

Prog City > add F george wallet spencer

Prog City > show
Team F
        jack willis
        anne james
        jerry lee lewis
        george wallet spencer
Team B
        lucas tee warlow
        jack willow
Team D
        johnny crash

Prog City >
```

Output 9. Exemple on s'afegeixen alguns bombers i es mostra el resultat la comanda "show".

2.4 Comanda “operation”

La comanda “operation” permet assignar directament una operació d'extinció d'incendi a un equip. La comanda segueix el següent format:

```
operation <id equip> <rang de temps>
```

L'identificador de l'equip consisteix en la lletra majúscula que el defineix. De la mateixa manera que succeïa en la comanda “add”, si l'equip no existeix s'ha de mostrar el mateix error que anteriorment (veure Output 7).

En cas que l'equip existeixi s'ha d'assignar aquesta operació al membre de l'equip en qüestió que porti menys minuts acumulats de treball (en cas d'empat, com a l'inici del programa, l'assignació segueix l'ordre en el qual s'han introduït les dades). El rang de temps de cada operació permet calcular els minuts que s'assignen a aquest bomber i el seu format és el següent:

```
<hora inicial-hora final>
```

Les hores estaran definides en el format 24h i aniran de les 0:00 a les 23:59. En cas que alguna de les hores no sigui correcta s'haurà de mostrar l'error corresponent (veure Output 10).

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > operation F 21:67-23:15
(ERROR) Wrong time format

Prog City >
```

Output 10. El format de les hores no és correcte.

Així, per exemple, un rang de temps definit per “15:55-20:10” comprèn un total de 255 minuts que és el valor que s'hauria d'emmagatzemar per al bomber triat (es pot observar que en l'estructura emmagatzemem únicament els minuts). També cal tenir en compte, com succeïa en la primera pràctica, que en cas que l'hora inicial sigui superior a l'hora final llavors es dedueix que el rang inclou tota la nit. Per exemple, el rang “23:45-10:04” hauria de resultar en un total de 619 minuts.

Una vegada assignada l'operació a l'equip i bomber, el programa quedarà a l'espera de la següent ordre (veure Output 11). Es pot suposar que el nombre d'operacions d'un bomber mai serà superior que MAX_OPERATIONS.

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > operation F 22:15-5:27

Prog City > operation K 10:28-11:17
(ERROR) Team K does not exist

Prog City > operation B 10:15-13:29

Prog City >
```

Output 11. S'introdueixen 3 operacions en els equips, una de las quals es descarta per ser incorrecta.

2.5 Comanda “show operations”

Amb la comanda “show operations” es podrà veure el detall de les operacions dels membres d'un equip. Com es pot observar aquesta comanda és una variació de la primera comanda “show”. Així, si la comanda rebuda és únicament “show” s'executarà la comanda indicada en l'apartat 2.2, mentre que si la comanda “show” va acompanyada de la paraula “operations” s'executarà la comanda que s'explica en aquest apartat. En cas de rebre una comanda “show” acompanyada d'una altra paraula diferent, s'actuarà com s'indica per a qualsevol comanda incorrecta i el programa no realitzarà cap acció (veure apartat 2).

La comanda “show operations” pot tenir dos possibles formats:

```
show operations <id equip>
show operations all
```

Com es pot observar, com a última dada la comanda pot rebre o bé l'identificador de l'equip que es vol mostrar o bé la paraula “all”. L'identificador de l'equip correspon a la lletra majúscula de l'equip. Si s'entra un únic caràcter i no correspon amb un equip vàlid, s'ha de mostrar el mateix error que ja s'ha indicat anteriorment (veure Output 7). Si s'introdueix la paraula “all” llavors la comanda fa referència a la totalitat dels equips. Si s'introdueix una paraula diferent de “all” com a última dada, llavors es considerarà que la comanda és incorrecte i no es realitzarà cap acció (veure apartat 2). En l'Output 12 es mostra un exemple amb tots dos errors.

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > operation B 10:15-13:29

Prog City > show operations hello

Prog City > show operations E
(ERROR) Team E does not exist

Prog City >
```

Output 12. Dos usos incorrectes de la comanda “show operations”.

Una vegada la comanda sigui correcta, es mostrarà per pantalla el detall de les operacions de l'equip en qüestió (o de tots), indicant per a cada membre el nombre d'operacions assignat i la dedicació de temps de cada operació (veure Output 13 i Output 14). Per a cada membre de l'equip i conjunt d'operacions, hi ha un tabulador (\t) en la sortida per pantalla.

```
Welcome to Prog City!

Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > operation F 10:23-17:16

Prog City > operation F 23:40-2:22

Prog City > operation F 11:15-18:15

Prog City > operation D 9:27-11:19

Prog City > show operations F
Team F
        jack willis: 1 operations
        [413]
        anne james: 2 operations
        [162 420]

Prog City >
```

Output 13. S'utilitza la comanda "show operations" per consultar les operacions de l'equip F.

```
Initial info: F/2/jack willis/anne james/B/1/lucas tee warlow/D/1/johnny crash.

Prog City > operation F 9:23-9:45

Prog City > operation D 11:25-6:33

Prog City > operation F 2:55-3:04

Prog City > operation D 3:35-5:55

Prog City > operation D 19:10-22:18

Prog City > show operations all
Team F
        jack willis: 1 operations
        [22]
        anne james: 1 operations
        [9]
Team B
        lucas tee warlow: 0 operations
        []
Team D
        johnny crash: 3 operations
        [1148 140 188]

Prog City >
```

Output 14. S'utilitza la comanda "show operations" per consultar les operacions de tots els equips.

2.6 Comanda “exit”

La comanda “exit” finalitza l'execució del programa, mostrant un missatge de comiat a l'usuari (veure Output 15).

```
Prog City > exit

See you later!
```

Output 15. La comanda “exit” finalitza l’execució del programa.

3. Consideracions

Per a la implementació de la pràctica cal tenir en compte les següents consideracions:

1. S'ha de seguir el format mostrat en els exemples.
2. En les sortides per pantalla es deixarà com a màxim una línia en blanc (observeu els exemples al llarg de l'enunciat).
3. La manera de gestionar els errors és estrictament la que demana l'enunciat. Si es diu que ha de tornar a introduir-se la dada no s'acceptarà que per exemple torni al menú inicial.
4. El codi ha d'estar correctament comentat de manera que sigui llegible sense dificultat. Comproveu que segueix la guia d'estils de l'assignatura.
5. El codi ha d'estar correctament estructurat en procediments i/o funcions. Es demana un mínim de 8 funcions o procediments (sense comptar el main) durant el desenvolupament de la pràctica. Es valorarà que l'objectiu, disseny i ús d'aquestes funcions/procediments siguin correctes (significança de les funcions o procediments, reaprofitament de codi, pas de paràmetres, etc).
6. Perquè la pràctica sigui acceptada és necessari que superi satisfactòriament tots els tests de CodeRunner i tenir una nota de Qualitat del Software superior a 1.
7. El programa hauria de desenvolupar-se íntegrament en l'entorn Linux de Matagalls, utilitzant l'editor vim per a escriure el codi C i el compilador gcc per a l'obtenció de l'executable corresponent. Aquí la podeu provar abans de testar-la al CodeRunner.
8. No es pot usar cap instrucció de C que no s'hagi explicat en classe.

4. Dates d'entrega i avaluació

La data de lliurament d'aquesta pràctica per a poder obtenir la màxima qualificació és el 15 de gener de 2023.

Perquè la pràctica es consideri **entregada** s'hauran de complir les següents condicions:

1. Executar el codi en el CodeRunner habilitat expressament i superar satisfactòriament els tests.
2. Entregar en el pou corresponent un arxiu .c amb el codi de la pràctica.
3. Entregar una memòria final de les pràctiques d'aquest semestre, incloent:
    a. Portada i índex
    b. Breu resum de l'enunciat de cada pràctica
    c. Diagrama d'activitats de la pràctica 1.1
    d. Estructuració (procs/funcs) de les pràctiques 1.2 i 1.3. Explicar la seva finalitat
    e. Principals dificultats i solucions aplicades
    f. Conclusions finals
    g. Estimació del temps dedicat per a realitzar cada pràctica

Es recorda que la pràctica serà avaluada de la següent manera:

- Execució: Té un pes del 80% i avalua el correcte funcionament de la pràctica. Serà la qualificació obtinguda en els tests de CodeRunner.

- Qualitat del SW: Té un pes del 20% i s'avaluarà la qualitat del codi lliurat i el seguiment de la Guia d'Estil de Programació que teniu disponible en el estudy.

L'avaluació de la pràctica serà sobre 10 punts, 8 punts o 6 punts, en funció de quan es lliuri la pràctica (i sigui acceptada):

Sobre 10 punts:	fins al 15 de gener de 2023, a les 23:59h.
Sobre 8 punts:	des del 16 de gener al 4 de juny de 2023, a les 23:59h.
Sobre 6 punts:	des del 5 de juny de 2023 al 2 de juliol de 2023, a les 23:59h.

Es recorda que la memòria té un pes d'un 10% en la nota final de pràctiques del semestre. Una nota de la memòria igual a 1 implica que la memòria no ha estat acceptada.