

Disseny descendent

Accions i funcions

Programació 1 – Grau en EI

Exercici 1

Donat un text acabat en ‘.’ que únicament conté paraules separades per blancs, implementeu un programa, emprant la tècnica del disney descendent d’algorismes, que compti el nombre de paraules del text. Les paraules seran de llargada màxima 20 i no es pot considerar cap restricció sobre el nombre absolut de paraules del text.

Exercici 2

Donat un text acabat en ‘.’ que únicament conté paraules separades per blancs, implementeu un programa, emprant la tècnica del disney descendent d’algorismes, que determini quants cops es repeteix la primera paraula dins el text. Les paraules seran de llargada màxima 20 i no es pot considerar cap restricció sobre el nombre absolut de paraules del text.

Exercici 3

Donat un text acabat en ‘.’ que únicament conté paraules separades per blancs, implementeu un programa, emprant la tècnica del disney descendent d’algorismes, que determini quantes paraules conté el text per a cada llargada. Les paraules seran de llargada mínima 1 i llargada màxima 20 i no es pot considerar cap restricció sobre el nombre absolut de paraules del text.

Exercici 4

Donat un text acabat en ‘.’ que únicament conté paraules separades per blancs, implementeu un programa, emprant la tècnica del disney descendent d’algorismes, que determini quants cops es repeteix cada paraula dins el text. Les paraules seran de llargada màxima 20, el text conté com a màxim 100 paraules diferents i no es pot considerar cap restricció sobre el nombre absolut de paraules del text.

Exercici 5

Donat un text acabat en '.' que únicament conté paraules separades per blancs, implementeu un programa, emprant la tècnica del disney descendent d'algorismes, que justifiqui les paraules del text en línies de 100 caràcters de manera que no es divideixin paraules entre línies. Les paraules seran de llargada màxima 20. Per fer la justificació eliminarem els blancs redundants existents al text i distribuïrem de forma uniforme nous caràcters blancs entre les paraules de cada línia. Evidentment, les línies justificades ni comencen ni acaben amb caràcters blancs.

Exercici 6

Donat un text acabat en '.' que únicament conté paraules separades per blancs, implementeu un programa, emprant la tècnica del disney descendent d'algorismes, que centri les paraules del text en línies de 100 caràcters de manera que no es divideixin paraules entre línies. Les paraules seran de llargada màxima 20. Per centrar les línies eliminarem els blancs redundants existents al text i distribuïrem de forma uniforme nous caràcters blancs entre l'inici i el final de cada línia.

Exercici 7

Donades dues seqüències d'enters de llargada màxima 100 acabades en 0 i ordenades creixentment, implementeu un programa, emprant la tècnica del disney descendent d'algorismes, que composi una única seqüència ordenada creixentment i que contingui tots els valors enters de les dues seqüències d'entrada. Tant les seqüències d'entrada com la de sortida podran tenir valors duplicats.

Per exemple, a partir de les seqüències de valors enters següents:

```
1 5 7 8 8 16 19 0
-5 4 6 7 9 15 17 45 0
```

La sortida és la següent:

```
-5 1 4 5 6 7 7 8 8 9 15 16 17 19 45
```

Exercici 8

Implementeu una funció, la qual anomenareu **discordancies**, que calculi el nombre de caràcters discordants (posició per posició) entre dues constants de cadena marcades amb el caràcter NUL '\0' (strings). Dos caràcters són discordants si són diferents.

Per exemple, la crida a la funció **discordancies** següent:

```
discordancies("dijous","diumenge");
```

retornarà el valor enter 6.

Exercici 9

Implementeu la funció de la biblioteca estàndar **strrchr** declarada al fitxer de capçalera `<string.h>`. El prototipus de la funció és el següent:

```
int strrchr(char s[ ], char c);
```

La funció **strrchr** cerca el caràcter **c** a la cadena **s** la qual està marcada amb el caràcter NUL `'\0'`. Si troba el caràcter retorna la posició de la darrera ocurrència del caràcter **c** a la cadena **s** si no, retorna el valor 0.

Per exemple, la crida **strchr**(`'MARIA'`, `'P'`) retorna 0 mentre que **strchr**(`'MARIA'`, `'A'`) retorna 5 ja que el caràcter 'A' ocorre en `'MARIA'` com a 2n i 5è caràcter.

Exercici 10

Implementeu la funció de la biblioteca estàndar de ANSI C **strcat**. El prototipus de la funció està definit al fitxer capçalera `<string.h>` i és el següent:

```
int strcat (char cs[ ], char ct[ ]);
```

La funció **strcat** concatena la cadena **ct** al final de la cadena **cs** i retorna la llargada de la cadena resultant. Ambdos cadenes estan marcades amb el caràcter NUL `'\0'`.

Exercici 11

Implementeu la funció de la biblioteca estàndar de ANSI C **strncat**. El prototipus de la funció està definit al fitxer capçalera `<string.h>` i és el següent:

```
int strncat (char cs[ ], char ct[ ], int n);
```

La funció **strncat** concatena **n** caràcters de la cadena **ct** al final de la cadena **cs** i retorna la llargada de la cadena resultant. Ambdos cadenes estan marcades amb el caràcter NUL `'\0'`.

Exercici 12

Implementeu la funció de la biblioteca estàndar de ANSI C **strlen**. El prototipus de la funció està definit al fitxer capçalera `<string.h>` i és el següent:

```
int strlen (char cs[ ]);
```

La funció **strlen** retorna la llargada de la cadena **cs** (nombre absolut de caràcters que componen la cadena **cs**). La cadena **cs** està marcada amb el caràcter NUL `'\0'`.

Exercici 13

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una matriu de valors enters de dimensió $F \times C$, i dos enters f i d . El programa mostrarà la matriu que resulta de desplaçar els valors de la fila f d posicions a l'esquerra amb realimentació per la dreta. Doneu un missatge d'error per al valor de fila fora de rang. Amb el desplaçament treballem mòdul el nombre de columnes.

Exercici 14

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una estructura tridimensional de valors enters de dimensió $P \times F \times C$ i calculi la suma de tota l'estructura.

Exercici 15

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una estructura tridimensional de valors enters de dimensió $P \times F \times C$ i calculi i calculi el valor i els índex del valor mínim.

Exercici 16

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una sopa de lletres de $N \times N$ i una paraula de mida màxima N acabada en punt i determini si la paraula està en alguna fila de la sopa de lletres (esquerra a dreta).

Exercici 17

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una sopa de lletres de $N \times N$ i una paraula de mida màxima N acabada en punt i determini si la paraula està en alguna columna de la sopa de lletres (dalt a baix).

Exercici 18

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una sopa de lletres de $N \times N$ i una paraula de mida màxima N acabada en punt i determini si la paraula està en alguna de les diagonals principals de la sopa de lletres (dalt a baix).

Exercici 19

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una sopa de lletres de $N \times N$ i una paraula de mida màxima N acabada en punt i determini si la paraula està en alguna de les diagonals de la sopa de lletres (dalt a baix).

Exercici 20

Repetiu els problemes anteriors (del 15 al 19) considerant les direccions que falten.

Exercici 21

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat dues matrius de valors enters amb dimensions $N \times M$ i $M \times P$, respectivament, i generi com a sortida la matriu que resulta del producte de matrius (matriu producte de dimensió $N \times P$).

Exercici 22

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una matriu de valors enters de dimensió $F \times C$, i dos enters $f1$ i $f2$. El programa mostrarà la matriu que resulta d'intercanviar els valors de la fila $f1$ amb els valors de la fila $f2$. Doneu un missatge d'error per als valors de fila fora de rang.

Exercici 23

Utilitzant la tècnica de disseny descendent dissenyeu i implementeu un programa que llegeixi de teclat una matriu de valors enters de dimensió $F \times C$, i dos enters $c1$ i $c2$. El programa mostrarà la matriu que resulta d'intercanviar els valors de la columna $c1$ amb els valors de la columna $c2$. Doneu un missatge d'error per als valors de columna fora de rang.

Exercici 24

Utilitzant la tècnica del disseny descendent d'algorismes, implementeu un programa que llegeixi de teclat un valor enter en notació decimal i mostri la representació binària i octal del valor introduït. Implementeu una funció parametritzada on la base de transformació també sigui un paràmetre d'entrada. No es permet utilitzar les operacions de conversió de llibreria. Considereu conversions amb un màxim de 20 dígit.

Exercici 25

Utilitzant la tècnica del disseny descendent d'algorismes, implementeu un programa que llegeixi de teclat una seqüència de valors enters acabada en 0 i mostri per pantalla la subseqüència més llarga ordenada creixentment. Considereu que la subseqüència més llarga contindrà com a màxim 50 valors.

Exercici 26

Utilitzant la tècnica del disseny descendent d'algorismes, implementeu un programa que llegeixi de teclat un valor enter en notació decimal i mostri la representació binària i octal del valor introduït. Implementeu una funció parametritzada on la base de transformació també sigui un paràmetre d'entrada. No es permet utilitzar les operacions de conversió de llibreria. Considereu conversions amb un màxim de 20 dígit.

Exercici 27

Utilitzant la tècnica del disney descendent d'algorismes, implementeu un programa que llegeixi de teclat una seqüència de valors enters acabada en 0 i determini si conté un valor igual a la suma de tots el previs. En cas afirmatiu, mostreu els valors que la componen. Considereu que la seqüència contindrà com a màxim 100 valors.

Exercici 28

Implementeu una acció que anomenareu `inversa` amb el prototipus següent:

```
void inversa (char c1[ ], char c2[ ]);
```

`inversa` emmagatzema en `c2` la inversa de la cadena emmagatzemada en `c1`. La cadena `c1` està marcada amb el caràcter NUL `'\0'`. La cadena `c2` quedarà marcada amb el caràcter NUL `'\0'`.

Exercici 29

Implementeu una acció que anomenareu `inversa_concat` amb el prototipus següent:

```
void inversa_concat (char c1[ ], char c2[ ], char c3[ ]);
```

`inversa_concat` emmagatzema en `c3` la inversa de la cadena emmagatzemada en `c1` amb la inversa de la cadena emmagatzemada en `c2`. Les cadenes `c1` i `c2` estaran marcades amb el caràcter NUL `'\0'`. La cadena `c3` quedarà marcada amb el caràcter NUL `'\0'`.

Exercici 30

Implementeu la funció de la biblioteca estàndar `strcmp` declarada al fitxer de capçalera `<string.h>`. El prototipus de la funció és el següent:

```
int strcmp(char s1[ ], char s2[ ]);
```

La funció `strcmp` retorna 0 si `s1` i `s2` són iguals, un negatiu si `s1` és menor que `s2` (ordenament lexicogràfic) i un negatiu si `s2` és menor que `s1` (ordenament lexicogràfic). Tant `s1` com `s2` estaran marcades amb el caràcter NUL `'\0'`.