

Programació C

Nom : _____

Primera part

Escriu les respostes en el lloc indicat amb una X. La qualificació d'aquesta primera part és 1/5 punt per resposta encertada, tenint present que si la resposta no és correcta hi ha un descompte de 1/20 punt. Al acabar, copia els resultats en el full de respostes, que **seran les úniques vàlides**.

1.- En un ordinador, s'anomena *software* a :

- *a) Al conjunt de programes que transformen les senyals en informació que les persones podem entendre
- *b) Tots els components no físics de l'ordinador
- *c) Al conjunt de programes que controlen els diferents dispositius de l'ordinador
- *d) Cap de les anteriors

2.- Els perifèrics són:

- *a) Els circuits que estan en la perifèria de l'ordinador
- *b) Dispositius que realitzen l'intercanvi de la informació entre l'ordinador i l'exterior.
- *c) Elements que transformen les magnituds físiques en senyals que poden ser processades per l'ordinador.
- *d) Elements que interpreten els codis de sortida de l'ordinador i els transformen en fenòmens apreciables per l'ésser humà.

3.- Quina de les següents afirmacions és correcta:

- *a) 100 Kb = 102.400 bytes.
- *b) 1 Mb = 1024 Kb.
- *c) 1 Gb = 102.400 Kb.
- *d) són correctes la a i la b.

4.- Un sistema operatiu és:

- *a) El conjunt de programes que sincronitzen els processos entre la CPU i les interrupcions.
- *b) El conjunt de programes que comuniquen entre si els diferents processos de control.
- *c) El conjunt de programes que actuen d'intermediari entre l'usuari i el *hardware* de l'ordinador
- *d) Cap de les anteriors.

5.- S'anomena *firmware*:

- *a) Al conjunt de programes i dades guardats en memòria ROM.
- *b) Al conjunt de programes i dades guardats en memòria RAM.
- *c) Al primer programa que executa la CPU quan comença el seu treball.
- *d) És correcta la a i la c

6.- Les sigles BIOS, volen dir:

- *a) Basic Installed Operating System.
- *b) Basic Input Operating System.
- *c) Base Interrupt Operating System.
- *d) Cap de les anteriors.

7.- L'interpret de comandes des DOS s'anomena:

- *a) IO.SYS
- *b) MSDOS.SYS
- *c) COMMAND.COM
- *d) IBMBIO.COM

8.- En el procés d'arranc de l'MSDOS:

- *a) Primer es busca el programa d'arranc anomenat BOOT primer en el disc flexible i després en el dur i es carrega en memòria.
- *b) Primer s'executen els programes de personalització de l'ordinador i després els de configuració del sistema.
- *c) Els programes IO.SYS i MSDOS.SYS es busquen i es carreguen abans que el COMMAND.COM
- *d) Són correctes la a i la c

9.- Quina de les següents afirmacions és correcta:

- *a) Els caràcters '\$' i '!' no es poden fer servir en l'identificador d'un fitxer.
- *b) Els caràcters '+' i '.' si que no es poden fer servir en l'identificador d'un fitxer.
- *c) Són correctes la a i la b
- *d) Cap de les anteriors.

10.- En el sistema operatiu DOS, els fitxers poden ser:

- *a) Ordinaris, directoris i especials.
- *b) Executables o de dades.
- *c) ordinaris, executables i de dades.
- *d) Cap de les anteriors.

11.- El sistema físic per emmagatzemar la informació en un ordinador és:

- *a) Discos durs i flexibles
- *b) Discos durs, flexibles i memòria RAM.
- *c) Sistema magnètic i òptic.
- *d) Cap de les anteriors.

12.- Per accedir a una informació en un ordinador s'utilitza:

- *a) Un sistema de dues coordenades (pista i sector)
- *b) Un sistema de dues coordenades(sector i cluster)
- *c) Un sistema de tres coordenades (pista, sector i cluster)
- *d) Cap de les anteriors.

13.- Tots els directoris excepte, l'arrel, tenen dues entrades:

- *a) La 1^a conté el nom de tots els fitxers creats que té el directori.
- *b) La 2^a conté el nom de tots els fitxers esborrats en el directori.
- *c) Les entrades es designen pels caràcters: '.' i '..'
- *d) Els directoris no tenen entrades, això són bajanades.

14.- Quina afirmació és correcta:

- *a) Els arxius de processament de text són fitxers amb format que porten l'extensió .BAT
- *b) Els fitxers executables han de tenir necessàriament l'extensió .EXE
- *c) Són correctes la a i la b
- *d) Cap de les anteriors

15.- Els atributs dels fitxers defineixen característiques pròpies:

- *a) L'atribut de lectura impedeix modificar les dades actuals però si afegir-ne de noves
- *b) L'atribut d'amagat es designa pel caràcter 'O' (Oculto)
- *c) Quan es desactiva l'atribut de lectura, es designa amb el caràcter 'W' (WRITE: es permesa l'escriptura)
- *d) Cap de les anteriors

16.- Quina afirmació és correcta:

- *a) Les comandes internes del DOS resideixen en la memòria principal
- *b) Les comandes externes del DOS formen part del programa COMMAND.COM
- *c) Els paràmetres en les ordres s'utilitzen per modificar la forma en que una ordre realitza una tasca
- *d) Són correctes la a i la c

17.- Quina afirmació és correcta:

- *a) Les ordres 'TYPE', 'COPY', i 'DEL' fan referència només a fitxers.
- *b) Les ordres 'DIR', 'MD' i 'MOVE' fan referència només a directoris.
- *c) L'ordre 'MOVE' fa referència tant a fitxers com a directoris
- *d) Són correctes la a com la c.

18.- En el sistema MSDOS, s'obren tres fitxers estàndards a disposició del programa anomenats: *stdin*, *stdout* i *stderr*. Per defecte aquests fitxers estan associats:

- *a) *stdin* al teclat, *stdout* i *stderr* al monitor.
- *b) *stdin* al teclat, *stdout* al monitor i *stderr* al disc.
- *c) *stdin* al teclat, *stdout* al disc i *stderr* al monitor.
- *d) Cap de les anteriors.

19.- Quin resultat tindrà l'ordre DIR > LLISTA.TXT

- *a) Busca en el directori actual el fitxer 'llista.txt' i si el troba escriu a pantalla les seves característiques. nom, longitud, hora i data de l'ultima modificació...
- *b) Guarda en el fitxer 'llista.txt' el nom i totes les característiques dels fitxers del directori actual.
- *c) Fa un 'DIR' a pantalla de tots els fitxers del directori 'LLISTA.TXT'
- *d) No fa res, perquè el caràcter '>' és il·legal.

20) Quina de les comandes següents és d'ús exclusiu en el programa 'CONFIG.SYS'

- *a) REM
- *b) BREAK
- *c) COUNTRY
- *d) PAUSE

21.- L'àrea de memòria superior és:

- *a) La memòria física que es col·loca a partir d'1Mb.
- *b) L'única que pot utilitzar el DOS.
- *c) La compresa entre 640 Kb i 1Mb.
- *d) Més lenta que la convencional.

22.- El programa que permet accedir a l'àrea de memòria superior i utilitzar de manera simulada tant la memòria estesa com l'expandida s'anomena:

- *a) HIMEM.SYS
- *b) EMM386.EXE
- *c) HIMEN.sys
- *d) Cap de les anteriors.

23.- Quina afirmació és correcta:

- *a) *Multitasca* és la característica que permet mantenir diferents aplicacions obertes, encara que només una s'executa.
- *b) *Multiprogramació* és la capacitat del sistema per alternar entre diferents aplicacions que s'estan executant al mateix temps.
- *c) *Memòria virtual* significa que el sistema utilitza una part de la memòria RAM com una extensió del disc dur, amb la qual cosa es guanya en velocitat.
- *d) Cap de les anteriors.

24.- Un algorisme és:

- *a) Conjunt d'instruccions que s'executen repetidament.
- *b) Un conjunt de sentències sense salts ni desviacions de la seqüència lineal d'instruccions del programa.
- *c) La descripció d'una tasca per mitjà d'operacions bàsiques.
- *d) Un programa en codi màquina.

25.- El resultat de la fase d'edició del programa és:

- *a) Un fitxer de text.
- *b) Un fitxer amb l'extensió .OBJ
- *c) Un fitxer executable.
- *d) Cap de les anteriors

26.- El llenguatge C és:

- *a) Un llenguatge interpretat
- *b) Un llenguatge compilat
- *c) Un llenguatge interpretat i compilat alhora.
- *d) Cap de les anteriors.

27.- El programa principal en el llenguatge C s'anomena:

- *a) Main
- *b) Mòdul arrel
- *c) Mòdul principal
- *d) Les tres afirmacions són vàlides.

28.- Quina afirmació és correcta:

- *a) La programació estructurada també s'anomena TOP-DOWN
- *b) La programació modular conté tres tipus d'estructures de control: seqüencials, condicionals i repetitives.
- *c) Cada mòdul d'un programa té un únic punt d'entrada i un altre de sortida.
- *d) Són correctes la b i la c.

29.- Quina afirmació és correcta:

- *a) Una dada de tipus alfanumèric està formada per un sol caràcter de l'alfabet o numèric.
- *b) Una dada lògica o booleana pot prendre tres valors: cert, fals, i no definit.
- *c) Una dada de tipus caràcter no pot contenir els caràcters especials: '+', '*', '-'
- *d) Cap de les anteriors

30.- A la representació gràfica d'una operació s'anomena diagrama de flux.

- *a) Si el diagrama és del sistema s'anomena: ordinograma
- *b) Si el diagrama és del procés s'anomena: organigrama
- *c) Si el diagrama és del sistema s'anomena: organigrama
- *d) Cap de les anteriors

RESPOSTES

1	7	13	19	25
2	8	14	20	26
3	9	15	21	27
4	10	16	22	28
5	11	17	23	29
6	12	18	24	30

Segona part

1r.- Canvia el nom de l'utilitat DELTREE.EXE i realitza un fitxer anomenat DELTREE.BAT que necessiti per funcionar dos arguments. El primer serà una paraula de pas (contrasenya), i el segon la unitat i la ruta del directori que volem esborrar. Per exemple si la paraula de pas és : 'X36' i el directori que volem esborrar 'C:\PROVA\UTIL', per executar el fitxer de processament per lots hauríem d'escriure:

DELTREE X36 C:\PROVA\UTIL

(2 p)

2n.- Dissenya un diagrama de flux que calculi la suma de n termes de la sèrie: 1, 2, 4, 8, 16,...

(2 p)

Solució de la primera part:

RESPOSTES

1	B	7	C	13	C	19	B	25	A
2	B	8	D	14	D	20	C	26	B
3	D	9	D	15	D	21	C	27	D
4	C	10	A	16	A	22	B	28	C
5	D	11	C	17	D	23	D	29	D
6	B	12	D	18	A	24	C	30	C

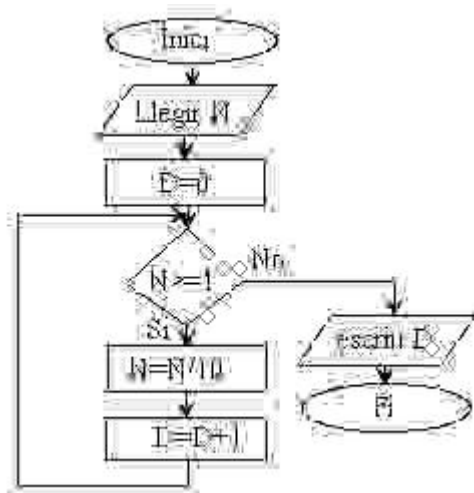
1a PART**Exercici 1 [2p]**

El pseudocodi següent correspon al sistema d'ordenació *'shell'*. Aquest sistema consisteix en comparar els elements separats per una **distància** de comparació que normalment és la meitat de la llista. En cada etapa o fase la **distància** es va dividint per dos fins que sigui menor que 1, quan sigui així la llista quedarà ordenada. La comparació entre els elements separats per la **distància** s'ha de seguir fent mentre hi hagi canvis en l'etapa.

```

inici
    dist=N/2
    mentre dist >= 1
        sw = 1
        mentre sw diferent 0
            sw = 0
            i = 1
            mentre i <= (N-dist)
                si llista[i] > llista[i+dist]
                    llavors
                        canvia elements
                        sw = 1
                fi_si
                i = i + 1
            fi_mentre
        fi_mentre
        dist = dist / 2
    fi_mentre
fi
  
```

- Dibuixa l'ordinograma corresponent a aquest pseudocodi.
- Aquest mètode d'ordenació serà més ràpid que el d'inserció? raona la resposta.

Exercici 2 [2p]

En l'ordinograma següent el valor de N pot ser un número comprés entre 1 i 50.000.

- Escriu el pseudocodi corresponent.
- Suposa per a N tres o quatre valors diferents. Quan valdrà D en cada cas?, quina relació hi ha entre N i D?

2a PART

Exercici 3 [3p]

Tenim un vector numèric d'una dimensió amb N elements. Cada posició del vector conté un número del 0 al 9, fent servir estructures del tipus : **mentre....**

- a) Dibuixa l'ordinograma d'un algoritme que permeti determinar si el número del vector numèric és **capicua**.
- b) Escriu el pseudocodi de l'ordinograma anterior.
- c) Torna a fer l'apartat a) però amb estructures del tipus: **repeteix..... mentre**

Exercici 4 [3p]

Tenim la llista numèrica següent:

POS 0	POS 1	POS 2	POS 3	POS 4	POS 5	POS 6	POS 7	POS 8	POS 9	POS 10	POS 11	POS 12	POS 13	POS 14	POS 15
0	2	4	8	10	14	23	25	26	29	32	34	38	43	49	57

Volem fer una recerca binària d'un element qualsevol.

- a) Explica breument en que consisteix aquest algoritme de recerca
- b) Dibuixa l'ordinograma corresponent a l'algoritme
- c) Omple la taula següent suposant que volem buscar el número 37.

LLEGENDA	E	N	I	S	C	Trobat
I : Índex inferior						
S : Índex superior						
N : Nombre d'elements						
C : Element central						

exercici 1 [2p]

Un programa conté les sentències següents:

```
#include <stdio.h>
char frase[80];
```

Escriu un programa que llegeixi caràcters d'un amb un amb l'ordre **getche()** i els emmagatzemi en la variable '**frase**' fins a trobar el caràcter '.' o bé omplir la cadena de caràcters.

exercici 2 [3p]

Escriu l'ordinograma d'un programa que demani a l'usuari un codi format per 4 lletres o números i només permeti continuar si els quatre caràcters entrats corresponen amb els guardats a la variable '**password**'. La comparació dels caràcters correctes s'ha de fer un a un. (No es pot comparar paraules senceres).

Tradueix el programa a C.

exercici 4 [2p]

Crea un programa per sumar dos números enters fent servir una funció per llegir les dades i una altre per a realitzar la suma.

exercici 4 [3p]

Fes un programa que preguntí a l'usuari si vol escriure només amb '**MAJÚSCULES**' o només amb '**minúscules**'. A continuació, anar llegint caràcters amb la funció **getch()** tot escrivint en la pantalla segons la preferència triada fins trobar el caràcter '* '.

Caràcters ASCII		Decimal	
A	Z	65	90
a	z	97	122

Solucions al examen del parcial del 2n trimestre

```
// Exercici 1
#include <stdio.h>
#include <conio.h>
void main()
{
    char frase[80];
    int n=0;
    clrscr();
    while (n<80&&frase[n-1]!='.') {
        frase[n]=getche();
        n=n+1;
    }
    frase[n]='\0';
    printf("%s",frase);
    getch();
}

// Exercici 2
#include <stdio.h>
#include <conio.h>
void main()
{
    char password[]="1234";
    char codi[4];
    int n,correcte;
    do {
        clrscr();
        printf("\n\n\n\t\tEntra la paraula d'accés : ");
        n=0;
        correcte=1;
        do {
            codi[n]=getch();
            putchar('*');
            if (correcte) if (codi[n]!=password[n]) correcte=0;
            n=n+1;
        } while (n<4);
        if (!correcte) {
            printf("\n\n\n\tCodi NO CORRECTE, torna-ho a provar");
            getch();
        }
    } while (!correcte);
    printf("\n\n\n\tCodi CORRECTE! ");
    getch();
}

// Exercici 3
#include <stdio.h>
#include <conio.h>
void llegir(void);
int suma(int a,int b);
int n1,n2;
void main()
{
    llegir();
    printf("\n\n\tEl resultat de la suma és : %d",suma(n1,n2));
    getch();
}
```

```

void llegir(void)
{
    clrscr();
    printf("\n\n\tEntra el 1r numero :");
    scanf("%d",&n1);
    printf("\n\n\tEntra el 2n numero :");
    scanf("%d",&n2);
}
int suma(int a,int b)
{
    return a+b;
}

// Exercici 4
#include <stdio.h>
#include <conio.h>
void main()
{
    char resp,car;
    do {
        clrscr();
        printf("\n\n\t1.- MAJUSCULES");
        printf("\n\n\t2.- minuscules");
        printf("\n\n\n\t\tTria una opció :");
        resp=getch();
    } while (resp!='1'&&resp!='2');
    clrscr();
    printf("\n\n\nEntra caràcters, per sortir prèmer '*' \n");
    do {
        car=getch();
        if (resp=='1') {
            if (car>96&&car<123) putchar(car-32);
            else putchar(car);
        }
        if (resp=='2') {
            if (car>64&&car<91) putchar(car+32);
            else putchar(car);
        }
    } while (car!='*');
    printf("\n\n\n\nPer sortir, prèmer una tecla.");
    getch();
}

```

Programació C

Nom : _____

Exercici 1 [4p]

Dissenya un programa que llegeixi text des del teclat fins trobar el caràcter '.' i al acabar escrigui a la pantalla el nombre de caràcters entrats així com el de paraules, per exemple en la frase:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
L	e	s		p	a	p	a	l	l	o	n	e	s			s	ó	n		f	r	à	g	i	l	s	.
1ª paraula				2 paraula								3ª paraula				4 paraula											

el programa hauria de dir: 28 caràcters i 4 paraules

Dibuixa l'**ordinograma** i **escriu el programa C** que correspongui.

Exercici 2 [2p]

En els programes següents digues quins són els respectius valors parcials de 'x' i de 'i' que s'escriuran en la pantalla, així com el seu valor final.

```
//ex_c5_2a.cpp
#include <stdio.h>
#include <conio.h>
void main()
{
    int i=0,x=0;
    clrscr();
    printf("\nEXERCICI 2(a)\n");
    do {
        if (i%5==0) {
            x++;
            printf("%d",x);
        }
        i++;
    } while (i<20);
    printf("\nx=%d",x);
    getch();
}
X=
X=
```

```
//ex_c5_2b.cpp
#include <stdio.h>
#include <conio.h>
void main()
{
    int i=0,x=0;
    clrscr();
    printf("\nEXERCICI 2(b)\n");
    for(i=1;i<10;i=i+2) {
        x++;
        printf("%d",i);
    }
    printf("\ni=%d",i);
    getch();
}
i=
i=
```

valors parcials
valors final

Exercici 3 [2p]

En una capsà hi ha 80 boles numerades del 1 al 80. Es treu una bola i el número s'apunta retornant la bola a la capsà. Aquest procés es repeteix fins que surti tres vegades el número 33. Fer un programa que escrigui en la pantalla quantes vegades surt cada bola.

La sortida del programa hauria de ser semblant a aquesta:

La bola 1 ha sortit 2 vegades.
La bola 2 ha sortit 1 vegades.
La bola 3 ha sortit 0 vegades.
La bola 4 ha sortit 4 vegades.
La bola 5 ha sortit 2 vegades.
La bola 6 ha sortit 1 vegades.
etc....

Exercici 4 [2p]

En matemàtiques es defineixen els nombres primers com aquells que només són divisibles per 1 o per si mateixos. El programa següent determina, donat un número, si és primer o no. Dibuixa l'ordinograma que correspongui a aquest programa.

```
//ex_c5_4.cpp
#include <stdio.h>
#include <conio.h>
void main ()
{
    int primer=1,n,i;
    printf("\nEntra el número : ");
    scanf("%d",&n);
    i=n-1;
    while (i>1 && primer==1) {
        if (n%i==0) primer=0;
        else i--;
    }
    if (primer==1) printf("\nÉs un número primer");
    else printf("\nNo és un número primer");
    printf("\n\n\tPer continuar, premer una tecla....");
    getch();
}
```

Solucions a l'examen final 2n trimestre

exercici 1

```
//ex_c5_1.cpp
#include <stdio.h>
#include <conio.h>
void main()
{
    int paraula,sc=0,sp=0;
    char car;
    clrscr();
    printf("\nEscriu un text : \n");
    paraula=1;
    do {
        car=getche();
        sc++;
        if(paraula && car!=' ') {
            sp++;
            paraula=0;
        }
        if(car==' ') paraula=1;
    } while(car!='. ');
    printf("\n\nEl text té %d caràcters i %d paraules",sc,sp);
    getch();
}
exercici 3
```

```
// ex_c5_3.cpp
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int main(void)
{
    int vector[81],n,y=1;
    clrscr();
    randomize();
    for(n=0;n<81;n++) vector[n]=0;
    do {
        n=random(80)+1;
        vector[n]=vector[n]+1;
    } while(vector[33]!=3);
    for(n=1;n<81;n++) {
        gotoxy(1,y);
        printf("La bola %d ha sortit %d vegades",n,vector[n]);
        gotoxy(40,y);
        n++;
        printf("La bola %d ha sortit %d vegades",n,vector[n]);
        y++;
        if(y>24) {
            printf("\nPrem una tecla per continuar.");
            getch();
            clrscr();
            y=1;
        }
    }
    getch();
    printf("\nPrem una tecla per continuar....");
    return 0;
}
```

Programació C

Nom : _____

Exercici 1 [2p]

Digues que fa el programa següent i dibuixa el seu ordinograma.

```
// ex_c6_1.cpp
#include <stdio.h>
#include <conio.h>
#define NF 20
void main(void)
{
    int c=40,f=1,n=1,a=0,i;
    clrscr();
    do {
        i=0;
        do {
                                gotoxy(c,f);
                                printf("*");
                                i++;
                                c++;
                            } while(i!=n);
        n=n+2;
        a++;
        c=40-a;
        f++;
    } while(f!=NF+1);
    getch();
}
```

Exercici 2 [2p]

Tenim un vector numèric (int), A de 10 files i 5 columnes i un altre B de 5 columnes i 10 files. Fes un programa que:

- Ompli totes les posicions del vector A amb números generats aleatòriament menors de 64 i majors de 32.
- Copia el contingut del vector A en B però invertint files per columnes. (la columna 3 del vector A s'ha de copiar en la fila 3 del vector B).

Exercici 3 [3p]

Escriu un programa que simuli realitzar una travessa de 15 resultats. El programa ha d'escriure els tres possibles resultats: '1', 'X', i '2' aleatòriament, tenint però en compte que la probabilitat que surti un '1' ha de ser del 50%, que surti una 'X' del 30%, i que surti un '2' del 20%.

Exercici 4 [3p]

Fes un programa que demani un text (màx. 80 caràcters) i el guardi en una cadena.

- Assigna a una variable punter, l'adreça del primer element de la cadena i escriu el text en la pantalla, caràcter a caràcter fent servir la variable punter.
- A la mateixa variable punter assigna-hi l'adreça de l'últim element i torna a escriure el text, ara però del revés.
- Escriu a la pantalla l'adreça del 1r. I del últim element de la cadena.

Nota: Per saber la longitud de la cadena pots fer servir la funció **strlen()**, que es troba en **string.h**

Solucions del exercicis

2.-

```
// ex_c6_2.cpp
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void main(void)
{
    int a[10][5],b[5][10],c,f;
    clrscr();
    randomize();
    for(f=0;f<10;f++) for(c=0;c<5;c++) a[f][c]=random(31)+33;
    for(f=0;f<5;f++) for(c=0;c<10;c++) b[f][c]=a[c][f];
    // escriu els vectors
    printf("\nVector A\n");
    for(f=0;f<10;f++) {
        for(c=0;c<5;c++) printf("%d\t",a[f][c]);
        printf("\n");
    }
    printf("\nVector B\n");
    for(f=0;f<5;f++) {
        for(c=0;c<10;c++) printf("%d\t",b[f][c]);
        printf("\n");
    }
    getch();
}
```

3.-

```
// ex_c6_3.cpp
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
void main(void)
{
    int i=0,p;
    char r;
    clrscr();
    randomize();
    while(i<15) {
        p=random(10)+1;
        if (p<6) r='1';
        else if(p>5 && p<9) r='X';
        else r='2';
        printf("\nPartit %d : %c",i+1,r);
        i++;
    }
    getch();
}
```

4.-

```
// ex_c6_4.cpp
#include <stdio.h>
#include <conio.h>
#include <string.h>
void main(void)
{
    char str[80],*p;
    int i=0;
    clrscr();
    printf("\nEntra una c...dena de text (m...x. 80 car):\n");
    gets(str);
}
```



```

printf("\n\n");
p=&str[0];
while(*p) {
    printf("%c", *p);
    p++;
}
p=&str[0]+strlen(str)-1;
printf("\n");
while(i<strlen(str)) {
    printf("%c", *p);
    p--;
    i++;
}
printf("\nL'adreça del primer element és : %X i la del l'últim
és %X", p, p+strlen(str)-1);
getch();
}

```

Exercici 1 [2p]

Fes un programa que conti i escrigui en la pantalla el número de bytes emmagatzemats en un fitxer, el nom del qual es passa al programa com a primer argument en la línia de comandes del DOS.

Exercici 2 [3p]

La funció **strcat()** afegeix el contingut d'una cadena en una altre, per exemple, la sentència:

```
strcat("Hola,"," com estàs");
```

afegeix la segona cadena a la primera que quedarà "Hola, com estàs".

Escriu un programa que fent servir una funció de nom **afegir()** faci exactament el mateix. El prototipus de la funció **afegir()** ha de ser:

```
void afegir(char *t1, char *t2);
```

Nota : No es pot fer servir la funció **strcat()**

Exercici 3 [2p]

Escriu un programa que llegeixi una línia pel teclat (fins trobar el senyal de 'return'), i escrigui tots els caràcters en un arxiu del disc "**inreves.dat**", un a un però invertint-los (les majúscules s'han de guardar en minúscula i les minúscules en majúscules). Es poden fer servir les funcions següents:

```
int islower(int ch);
```

Retorna diferent de zero si ch és una minúscula

```
int isupper(int ch);
```

Retorna diferent de zero si ch és una majúscula

```
int toupper(int ch);
```

Si ch és una minúscula, retorna majúscula, sinó retorna el mateix caràcter

```
int tolower(int ch);
```

Si ch és una majúscula, retorna minúscula, sinó retorna el mateix caràcter

na el mateix caràcter

Exercici 4 [3p]

Escriu un programa que utilitzi una funció **'recursiva'** per calcular i retornar la potència d'un número diferent de zero, essent el seu exponent enter i positiu.

```

Solucions al examen final
/* exercici nº1, examen ex_c7.doc */
// ex_c7_1.cpp
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

void main(int argc, char *argv[])
{
    FILE *arxiu;
    char nom[25];
    unsigned long l=0;
    clrscr();
    if(argc<2) {
        puts("Aquest programa conta el nombre de bytes d'un fitxer, ");
        puts(",s necessari el nom. Prem una tecla per sortir.");
        getch();
        exit(0);
    }
    if((arxiu=fopen(argv[1], "rb"))==NULL) {
        printf("\nNo hi ha acc,s a l'arxiu %s", argv[1]);
        exit(0);
    }
    fgetc(arxiu);
    while(!feof(arxiu)) {
        fgetc(arxiu);
        if(ferror(arxiu)) {
            printf("\nError al llegir l'arxiu %s", nom);
            exit(1);
        }
        l++;
    }
    fclose(arxiu);
    printf("\nL'arxiu '%s' t, %u bytes", argv[1], l);
    getch();
}

// soluci  exercici 2 de l'examen ex_c7.doc
// ex_c7_2.cpp
#include<stdio.h>
#include<conio.h>
main()
{
    void afegir(char *t1, char *t2);
    char text1[80], text2[80];
    clrscr();
    printf("\nEntra la frase 1 :");
    gets(text1);
    printf("\nEntra la frase 2 :");
    gets(text2);
    afegir(text1, text2);
    printf("\nLa frase afegida ,s :\n%s", text1);
    printf("\nPrem una tecla per continuar.");
    getch();
    return 0;
}

```

```

void afegir(char *t1, char *t2)
{
    /* Trobar el final de t1 */
    while(*t1) t1++;
    /* Afegir la cadena */
    while(*t2) *t1++=*t2++;
    /* Afegir al final el car...cter nul */
    *t1='\0';
}

/* exercici nº3, examen ex_c7.doc */
// ex_c7_3.cpp
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <ctype.h>
#include <string.h>
void main(void)
{
    char *linia;
    FILE *F1;
    int p=0;
    clrscr();
    printf("Entra una linia de text :\n");
    gets(linia);
    if((F1=fopen("inreves.dat","w"))==NULL) {
        printf("\nError en l'acc,s a l'arxiu");
        exit(0);
    }
    do {
        if(linia[p]!='\0') {
            if(islower(linia[p]))
                fputc(toupper(linia[p]),F1);
            else fputc(tolower(linia[p]),F1);
        }
        else fputc('\0',F1);
        p++;
    } while(p<=strlen(linia));
    fclose(F1);
}

/* exercici nº4, examen ex_c7.doc */
#include <stdio.h>
#include <conio.h>
float potencia(float x, int y);
void main(void)
{
    float base, poten;
    int exp;
    clrscr();
    printf("\n\nEntra la base\n");
    scanf("%f",&base);
    printf("\nEntra el exponent\n");
    scanf("%d",&exp);
    poten=potencia(base,exp);
    printf("\n\nEl valor de la potŠncia ,s : %0.2f ",poten);
    getch();
}

```

```
float potencia(float x, int n)
{
    float pot;
    if(n==0) pot=1;
    else pot=x*potencia(x, (n-1));
    return pot;
}
```