

1^{er} Parcial Programación I

Nombre:	Apellido	Legajo
N ^{ro} de hojas adicionales:		

1) [1 pt.] Indicar cuál es la salida del siguiente código:

```
int main (void)
{
    int r,p;
    p=(char) 0X12C;
    r=p++;
    printf ("%x\n",r) ;
}
```

Rta: **2c**

2) [1 pt.] ¿Cuál es la salida del siguiente programa?

```
#include<stdio.h>
#define MAXI 5
int a;
int b=3;

int main (void)
{
    while (a < MAXI && b-->=0)
        printf ("%d%d\n",a,b);
        a++;
        printf ("%d%d\n",a,b);
}
```

Respuesta:

a	b
0	2
0	1
0	0
0	-1
1	-2

3) [1 pt.] Decir cuál es la salida en pantalla del siguiente programa.

```
#include <stdio.h>
char mensaje[]="ABCD1234" ;
char *p;

int main (void)
{
    p= mensaje+3;

    printf("%c ",*--p);

    printf("%c ",*p++);

    printf("%c ",(*p)++);

    printf("%c \n\n",*p);

    printf("%s \n\n",mensaje);

    printf("%c ",*(--p));

    printf("%c ",*(p+1));

    printf("%c ",*mensaje);

    printf("%c \n\n",*mensaje + 1);

    while (*p++)
        printf("%c", *p);

}
```

Respuesta:

C C D E

ABCE1234

C E A B

E1234

4) [1 pt.] Indicar que problema presenta el siguiente código. Proponer alguna modificación que lo solucione.

```
int main (void)
{
    double x = 0.11;

    while(x -= 0.01)
        printf("%f\n", x);
}
```

No es correcto comparar con "!=" o "==" números en punto flotante pues la representación interna siempre tiene diferencias en las partes menos significativas. Lo correcto es definir un error máximo admisible. Este problema puede llevar a que el lazo en el programa anterior sea infinito pues puede ser que nunca se llegue a la condición de salida: $y = 0$ (¡exactamente 0!). Esto se puede arreglar, por ejemplo, así:

```
#include<stdio.h>

int main()
{
double x = 0.21;
while ((x -= 0.01)>0)
printf("%f\n",x);
}
```

5) [2 pts] Implementar las siguientes funciones

a) unsigned long factorial (int n)

La función calcula el factorial de un número natural.

b) int copiarn(char fuente[], char destino[], int n)

La función copia como máximo n caracteres del string fuente en el string destino y devuelve el número de caracteres copiados. A continuación escribir un programa (main) que haga uso de la misma.

6) [1 pt.] Escribir un programa que complete un arreglo con los números pares comprendidos entre 1 y 100 y los imprima en pantalla en orden descendente.

```
#include <stdio.h>

int main(void)
{
    int x,cuenta,z,i,arreglo[50];

    i=0;x=100;
    for (x=100;x>=1;x--)
    {
        cuenta=0;
        if (x%2==0)
        {
            arreglo[i]=x;
            i++;
        }
    }

    for (x=0;x<i;x++)
    {
        printf("%d\n",arreglo[x]);
    }

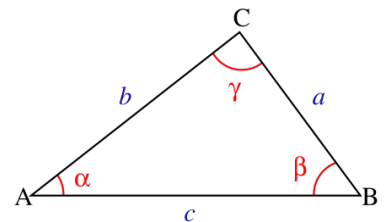
    return 0;
}
```

- 7) [2pts] La **fórmula de Herón**, atribuida a Herón de Alejandría , permite calcular el área de un triángulo en función de las longitudes de sus lados a, b y c:

$$\text{Área} = \sqrt{s(s-a)(s-b)(s-c)}$$

donde **s** es el semiperímetro del triángulo:

$$s = \frac{a + b + c}{2}.$$



Conociendo la función:

```
double getfloat (void)
```

Escribir un programa que solicite los valores de los lados a,b,c, calcule el área de un triángulo y lo presente en pantalla.

Los valores de los lados son números reales positivos.

La entrada de datos debe ser validada para que el triangulo sea posible. (Recordar que un lado debe ser menor que la suma y mayor que la diferencia de los otros dos lados).

```
//  
// main.c  
// heron  
//  
// Created by Ricardo Pingitore on 3/9/15.  
// Copyright (c) 2015 Ricardo Pingitore. All rights reserved.  
//  
  
#include <stdio.h>  
#include <math.h>  
  
double getfloat (void);  
  
int main(void) {  
    float a,b,c;  
    float s;  
    printf("Cálculo del área de un triángulo utilizando la Fórmula de He-  
rón\n");  
    printf("Ingrese el valor de los lados \n");  
    printf("a= ?\n");  
  
    a=getfloat();  
  
    printf("b= ?\n");  
    b=getfloat();  
  
    printf("c= ?\n");  
    c=getfloat();  
    s=(a+b+c)/2;  
  
    if(((a+b)<c)||((a+c)<b)||((b+c)<a))  
        printf("El triangulo no es posible");  
    else  
        printf("El área del triángulo es %f\n",sqrt((s*(s-a)*(s-b)*(s-c))));
```

```

    return 0;
}

double getfloat (void)
{
    double num=0,i;
    int c;
    int nega=0;
    if ((c=getchar()) == '-')
        nega=1;
    else
        num=(c-'0');
    for(;;(c=getchar()) != '\n';)
    {
        if ((c=='.') || (c==','))
            break;
        num=(num*10)+(c-'0');
    }
    if ((c=='.') || (c==','))
        for (i=10; ((c=getchar()) != '\n');i=(i*10))
        {
            num= (num+((c-'0')/i));
        }
    if( nega == 1 )
        num=(num)*-1;

    return num;
}

```

Ej 1	Ej 2	Ej 3	Ej 4	Ej 5	Ej 6	Ej 7	Total
1	1	1	1	2	2	2	10