

TÉCNICAS DE PROGRAMACIÓN
PRIMER LABORATORIO
SEMESTRE ACADÉMICO 2019-2

Horario: H0431

Duración: 110 minutos

Elaborado por los profesores del curso.

ADVERTENCIAS:

- Todo dispositivo electrónico (teléfono, tableta, computadora u otro) deberá permanecer apagado durante la evaluación.
- Coloque todo aquello que no sean útiles de uso autorizado durante la evaluación en la parte delantera del aula, por ejemplo, mochila, maletín, cartera o similar, y procure que contenga todas sus propiedades. La apropiada identificación de las pertenencias es su responsabilidad.
- Si se detecta omisión a los dos puntos anteriores, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- Es su responsabilidad tomar las precauciones necesarias para no requerir la utilización de servicios higiénicos: durante la evaluación, no podrá acceder a ellos, de tener alguna emergencia comunicárselo a su jefe de práctica.
- En caso de que el tipo de evaluación permita el uso de calculadoras, estas no podrán ser programables.
- Quienes deseen retirarse del aula y dar por concluida su evaluación no lo podrán hacer dentro de la primera mitad del tiempo de duración destinado a ella.

INDICACIONES:

- No se pueden usar apuntes de clase ni calculadoras.
- Está prohibido el acceso a Internet y a correo electrónico hasta que lo indiquen los jefes de práctica, tampoco podrá emplear dispositivos USB.
- Grabe el archivo en la dirección que los jefes de práctica indiquen, en la Intranet del curso. Use el formato de nombre de archivo que se le indique, caso contrario NO SE CORREGIRÁ su trabajo y se le asignará la nota cero (00).

PRIMERA PARTE: Uso del entorno NetBeans

En esta primera parte usted creará un proyecto con el nombre “Pregunta01”, escribirá el programa que se le proporciona, lo compilará, ejecutará, depurará y contestará a las preguntas que se le harán.

Se desea determinar el área bajo una curva y su longitud, de funciones del tipo:

$$y = a \cdot x^5 + b \cdot x^4 + c \cdot x^3 + d \cdot x^2 + e \cdot x + f$$

Por ejemplo para las curvas:

$$y_1 = 0.005878 \cdot x^5 - 0.00547 \cdot x^4 - 0.3654 \cdot x^3 - 0.9211 \cdot x^2 + 6.2587 \cdot x + 13.84$$

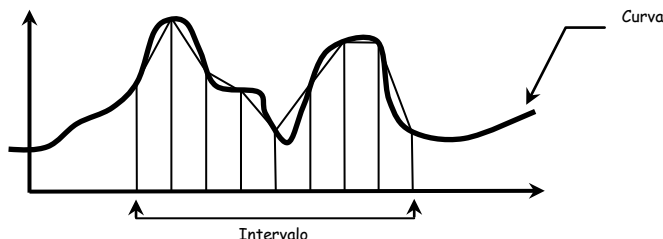
para el intervalo $[-4, 2]$

O

$$y_2 = 0.000541 \cdot x^5 - 0.0045127 \cdot x^4 + 0.06554 \cdot x^3 + 1.7557 \cdot x^2 - 0.8771 \cdot x + 5.245$$

para el intervalo $[-2, 1]$

El cálculo se hará para un intervalo dado, dividiendo la curva en segmentos del mismo ancho, sumando las áreas de los trapecios que se forman o los segmentos de recta entre los intervalos, como se muestra en la figura:



La determinación de los resultados se hará por aproximaciones sucesivas, esto es, primero se hará el cálculo con un solo intervalo, luego con dos, tres, etc. Cuando no haya diferencias entre dos cálculos sucesivos se dará por concluido el proceso.

El programa que resuelve este problema se muestra a continuación:

EN EL ARCHIVO main.c TENEMOS:

```
/*
 * Archivo: main.c
 * Autor:   ***COLOQUE AQUÍ SU CÓDIGO, NOMBRE Y APELLIDO ***
 * Fecha y Hora: ***COLOQUE AQUÍ LA FECHA Y HORA EN LA QUE EMPEZÓ A ESCRIBIR ***
 *
 * *****
 * Programa que calcula el área bajo la curva de un intervalo dado,
 * mediante aproximaciones sucesivas.
 * *****
 */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "funciones.h"
#define PRECISION 0.001

int main(int argc, char** argv) {
    double a, b, c, d, e, f;
    double limiteInf, limiteSup, h, x1, x2, y1, y2;
    int numIntervalos=0, intervalo;
    double area=0.0, areaAnt, longitud=0.0, longitudAnt;

    // Ingresamos los coeficientes de la ecuación:
    printf("Ingrese los coeficientes a, b, c, d, e, f: "),
    scanf("%lf %lf %lf %lf %lf %lf", &a, &b, &c, &d, &e, &f);

    /* Ingresamos los límites entre los que queremos calcular
       el área y la longitud de la curva:
    */
    printf("Ingrese los limites: ");
    scanf("%lf %lf", &limiteInf, &limiteSup);
    /* Calculamos el área y longitud de la curva para diferentes
       números de franjas
    */

    do{
        areaAnt = area;
        longitudAnt = longitud;
        area = longitud = 0;
        numIntervalos++;
        h = (limiteSup - limiteInf)/numIntervalos;
        x1 = limiteInf;
        for(intervalo=0; intervalo<numIntervalos; intervalo++){
            x2 = x1 + h;
            y1 = fx(a, b, c, d, e, f, x1);
            y2 = fy(a, b, c, d, e, f, x2);
            area += (y1+y2)*h/2;
            longitud += sqrt(potencia(y2-y1,2) + potencia(h,2));
            x1 = x2;
        }
        /* Salimos del while cuando no hay diferencia entre el
         * cálculo de dos áreas seguidas y el cálculo de dos
         * longitudes seguidas.
         */
    } while(fabs(area-areaAnt)>=PRECISION AND
           fabs(longitud-longitudAnt)>=PRECISION) ;

    // Mostramos los resultados:
    printf("Area bajo la curva   = %12.4lf\n", area);
    printf("Longitud de la curva = %12.4lf\n", longitud);

    return (EXIT_SUCCESS);
}
```

EN EL ARCHIVO funciones.h TENEMOS:

```
/*
 * Archivo:    funciones.h
 * Autor:     ***COLOQUE AQUÍ SU CÓDIGO, NOMBRE Y APELLIDO ***
 */

#ifndef FUNCIONES_H
#define FUNCIONES_H

    double fx(double, double, double, double, double, double, double);
    double potencia(double , int );
#endif /* FUNCIONES_H */
```

EN EL ARCHIVO funciones.c TENEMOS:

```
/*
 * Archivo:    funciones.c
 * Autor:     ***COLOQUE AQUÍ SU CÓDIGO, NOMBRE Y APELLIDO ***
 */

#include "funciones.h"

double fx(double a, double b, double c, double d, double e, double f, double x){
    double funcion;
    funcion = a*potencia(x,5) + b*potencia(x,4) + c*potencia(x,3) +
            d*potencia(x,2) + e*x + f;
    return funcion;
}

double potencia(double base, int exponente){
    int i;
    double solucion=1;
    for(i=0;i<exponente;i++){
        solucion=solucion*base;
    }
    return solucion;
}
```

Según lo anterior:

1. (3.5 puntos) Utilizando el entorno de desarrollo NetBeans, escriba el programa anterior (cópielo completamente tal como está en la hoja, incluyendo comentarios y, agregando lo que se indica.
2. (1.5 puntos) Verifique que ha escrito correctamente su programa, para esto compílelo y anote en las líneas siguientes los mensajes que le aparecen en la ventana de salida en su primera compilación:

.....
.....
.....
.....
.....

3. (1 puntos) Corrija y verifique que su programa se ejecuta correctamente, para eso, si ingresa los datos para y1 debe obtener como resultado 40.1810 para el área y 31.0734 para la longitud.
4. (1.5 puntos) Indique en la siguiente línea cuáles son los resultados para y2.

.....

5. Empleando las opciones de depuración del NetBeans indique lo siguiente:

(1 puntos) ¿Cuánto vale el área con 9 segmentos para y_1 ?:

(1.5 puntos) Indicar cuántos ciclos se requieren para calcular los resultados en y_2 :

.....

(*)

Nombre del Alumno

Código

Aula

(*) Debe entregar esta hoja, junto con el programa digitado en el entorno NetBeans, al final de la prueba.

SEGUNDA PARTE: Elaboración de un proyecto en NetBeans

Existen diversos algoritmos que permiten determinar si un número es primo o no, entre todos esos métodos, en el año 2002, Manindra Agrawal, Neeraj Kayal y Nitin Saxena, científicos del Instituto tecnológico hindú de Kanpur (India), desarrollaron un algoritmo predictivo capaz de determinar si un número es primo o no. Este algoritmo, fue bautizado como **Test de primalidad AKS**, por las iniciales de los apellidos de los investigadores que trabajaron en él.

El algoritmo se basa en la siguiente premisa: “para todo n , si $(x + 1)^n - (x^n + 1)$ es divisible entre n , entonces n será un número primo”.

Por ejemplo, para $n = 5$, tenemos que:

$$\begin{aligned}(x + 1)^5 - (x^5 + 1) &= x^5 + 5x^4 + 10x^3 + 10x^2 + 5x^1 + 1 - x^5 - 1 \\ &= 5x^4 + 10x^3 + 10x^2 + 5x^1\end{aligned}$$

Donde podemos apreciar que todos los coeficientes son divisibles entre 5, por lo que se puede afirmar que sea cual sea el valor de x , el resultado será múltiplo de 5. Por lo tanto 5 es primo.

Sin embargo, para $n = 4$, tenemos que:

$$\begin{aligned}(x + 1)^4 - (x^4 + 1) &= x^4 + 4x^3 + 6x^2 + 4x^1 + 1 - x^4 - 1 \\ &= 4x^3 + 6x^2 + 4x^1\end{aligned}$$

Encontramos aquí que 6 no es múltiplo de 4, por lo tanto 4 no es primo.

Una forma de poder determinar el valor de cada coeficiente es mediante la siguiente fórmula:

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Donde n es el exponente y r es la posición del coeficiente que se quiere evaluar. Así por ejemplo, para $n = 4$, tenemos que:

$$(x + 1)^4 = \overbrace{x^4}^{\uparrow 0} + \overbrace{4x^3}^{\uparrow 1} + \overbrace{6x^2}^{\uparrow 2} + \overbrace{4x^1}^{\uparrow 3} + \overbrace{1}^{\uparrow 4}$$

Posición (r) \rightarrow

Si queremos calcular el coeficiente para la posición 3: $\binom{4}{3} = \frac{4!}{3!(4-3)!} = \frac{4 \cdot 3 \cdot 2 \cdot 1}{3 \cdot 2 \cdot 1 \cdot (1)} = \frac{24}{6} = 4$

6. (10 puntos) Se pide escribir un programa en lenguaje C que permita leer un rango de números (a y b) y determine e imprima, empleando el algoritmo planteado anteriormente, los números primos que se encuentren en ese rango. El máximo valor permitido para b será 64. Debe validar que el rango ingresado sea correcto ($b > a$).

A continuación se presentan algunos ejemplos de ejecución:

```
Ingrese el rango a y b: 34 41
En ese rango son PRIMOS:
37
41
Total de valores primos: 2
-----
Ingrese el rango a y b: 34 88
EL RANGO DE VALORES NO PUEDE SOBREPASAR EL VALOR 64
-----
Ingrese el rango a y b: 34 18
EL RANGO DE VALORES NO NO ES CORRECTO
```

Si usted lo cree conveniente, la pregunta podrá ser elaborada empleando funciones, sin embargo, de hacerlo, éstas deberán ser desarrolladas obligatoriamente en archivos independientes (.h y .c) al archivo main.c.

CRITERIOS DE CALIFICACIÓN EN ESTA PREGUNTA:

1. Si el programa presentado que presenta más de tres errores de sintaxis serán calificados sobre la mitad del puntaje.
2. Si el programa no muestra los resultados o los muestren y no sean correctos, no podrán tener más del 75% de la nota.
3. Se descontará 15% de la nota si el programa define variables con nombres que no tengan sentido. Las variables deben empezar con una minúscula, se emplearán mayúsculas para separar las palabras compuestas (p. e.: baseInf).
4. Se descontará 15% de la nota si no se colocan comentarios relevantes, incluyendo un encabezado al inicio del programa en el que se indique el nombre del autor, la fecha, y una descripción de lo que hace programa).
5. No se calificará el código puesto como comentario.
6. No se calificarán aquellas funciones implementadas en el archivo main.c

San Miguel, 3 de septiembre del 2019