**Logic Specification Template**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student | Juan Alberto Gutierrez Canto | | Date | 14/06/2016 |
| Program | Normalizacion | | Program # | 9A |
| Instructor | Jorge Rafael Aguilar Cisneros | | Language | C++ |
| Object | N/A | | Function | ISEMPTY() |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | Using namespace std; | | |
|  | |  | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Si el nodo llamado raiz es igual a null |
|  | En caso que si regresar 1 |
|  | En caso que no regresar 0 |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | Struct nodo |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | Using namespace std; | | |
|  | | Double numero x | | |
|  | | Double numeroy | | |
|  | | Struct nodo\*sig | | |
|  | | Struct nodo raiz=NULL | | |
|  | | Struct nodo last=NULL | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Declaración de variables |
|  | El nodo llamado raíz es null |
|  | El nodo llamado last es null |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | INSERT() |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <malloc.h>  #include <studio.h>  Using namespace std; | | |
|  | | Struct nodo \*Nuevo=NULL | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Si la lista esta vacia |
|  | En caso que si |
|  | Reservamos memoria para el nodo |
|  | Guardamos valor de x |
|  | Guardamos valor de y |
|  | Apuntamos siguiente como nulo |
|  | Raíz es igual a last que es igual a nuevo |
|  | En caso de que no |
|  | Reservamos memoria para el nodo |
|  | Guardamos valor de x |
|  | Guardamos valor de y |
|  | Apuntamos siguiente como nulo |
|  | Apuntamos sig de last como nuevo |
|  | Last es igual a nuevo |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | REMUEVE |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <stdio.h>  #include <cstdlib>  Using namespace std; | | |
|  | | Struct nodo \*elimina=NULL | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Si la lista no esta vacia |
|  | En caso de que si |
|  | Elimina es igual a raíz |
|  | Raíz es igual a raíz que apunta en siguiente |
|  | Liberamos espacio de memoria de elimina |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | PEDIRDATOS |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <stdio.h>  Using namespace std; | | |
|  | | Double datox | | |
|  | | Double datoy | | |
|  | | Long long i | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Pedir n datos |
|  | Leer n datos |
| 2 | Inicializar i=0 |
|  | Mientras i<n |
|  | Pedir x & y |
|  | Leer x & y |
|  | Insertar en la lista ligada |
|  | I++ |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | BURBUJALTST |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <stdio.h>  Using namespace std; | | |
|  | | Struct nodo \*inicio | | |
|  | | Struct nodo \*siguiente | | |
|  | | Double datox | | |
|  | | Long long i | | |
|  | | Long long y | | |
|  | | Bool ambas=false | | |
|  | | Double datoy | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Inicialiar i=0 |
|  | Inicializar y=0 |
|  | Inicio es igual a raíz |
| 2 | Mientras i<n |
|  | Inicializar y=i+1 |
| 3 | Auxiliar es igual a lo que apunta incio en siguiente |
|  | Mientras y<n |
| 4 | Si ambas es falsa |
|  | En caso que si |
| 5 | Si datox de incio es mayor a datox de Auxiliar |
|  | En caso que si |
|  | Datox es igual a lo que hay en datox de Auxiliar |
|  | Datoy es igual a lo que hay en datoy de Auxiliar |
|  | Datox de Auxiliar es igual a lo que hay en datox de inicio |
|  | Datoy de Auxiliar es igual a lo que hay en datoy de inicio |
|  | Datox de inicio es igual a datox |
|  | Datoy de incio es igual a datoy |
|  | En caso de que no |
|  | Si datoy de incio es mayor a datoy de siguiente |
|  | En caso que si |
|  | Datox es igual a lo que hay en datox de Auxiliar |
|  | Datoy es igual a lo que hay en datoy de Auxiliar |
|  | Datox de Auxiliar es igual a lo que hay en datox de inicio |
|  | Datoy de Auxiliar es igual a lo que hay en datoy de inicio |
|  | Datox de inicio es igual a datox |
|  | Datoy de incio es igual a datoy |
|  | Y++ |
|  | Auxiliar es igual a lo que apunta Auxiliar en siguiente |
|  | I++ |
|  | Inicio es igual a lo que apunta incio en siguiente |
|  | Inicialiar i=0 |
|  | Inicio es igual a raíz |
| 6 | Mientras i<n |
|  | Imprime el valor de x & y |
|  | I++ |
|  | Inicio es igual a lo que apunta incio en siguiente |
| 7 | Si ambas es igual a false |
|  | En caso de que si |
|  | Ambas es igual a true |
|  | BURBUJALTST() |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | AVG |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | Using namespace std; | | |
|  | | Struct nodo \*inicio | | |
|  | | Double avgdata | | |
|  | | Int i | | |
|  | |  | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
|  | Inicio = raíz; |
|  | Avgdata=0; |
| 1 | Para i=0, mientras i<n, i++ |
|  | Avgdata+=numero x en inicio |
|  | Incio=sig en inicio; |
|  | Avgdata=avgdata/n |
|  | Return avgdata |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | VARIANZA |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <math.h>  Using namespace std; | | |
|  | | Struct nodo \*inicio | | |
|  | | Double varianza | | |
|  | | Int i | | |
|  | | Double avgdata | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
|  | Inicio = raíz; |
|  | Varianza=0; |
| 1 | Para i=0, mientras i<n, i++ |
|  | Varianza+=pow(numerox en inicio -avgdata,2) |
|  | Incio=inicio->sig; |
|  | varianza=varianza/n-1 |
|  | Return varianza |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | CALRANGOS |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | Using namespace std; | | |
|  | | Int n | | |
|  | | Int s | | |
|  | | Double tablavaloresnor | | |
|  | | Int i | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | If(n mod 5 == 0) |
| 2 | Para i=0, mientras i< n/5, i++ |
|  | Tablavaloresnor en posición 3,i=5; |
| 3 | Else if(n mod 5 == 4) |
| 4 | Para i=0,mientras i<n/5, i++ |
|  | Tablavaloresnor en pocicion 3,i=5 |
|  | Tablavaloresnor en pocicion 3,i=4 |
| 5 | Else if(n mod 5<4) |
| 6 | Para i=0,mientras i<n/5-1,i++ |
|  | Tablavaloresnor en pocicion 3,i = 5 |
| 7 | Para i=0 mientras i< n mod 5 + 5,i++ |
|  | Tablavaloresnor en posición 3, i+1 ++ |
| 8 | Return n/5-1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | REGLASMPNOR |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <math.h>  Using namespace std; | | |
|  | | Double xlow | | |
|  | | Double xhigh | | |
|  | | Double N | | |
|  | | Double W | | |
|  | | Double fija | | |
|  | | Double x2 | | |
|  | | Double ex | | |
|  | | Double fx | | |
|  | | Double term | | |
|  | | Double sumterm | | |
|  | | Bool negative | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
|  | Negative=false |
|  | Fija=1/sqrt(2\*pi) |
|  | N=20 |
| 1 | If(xhigh <0) |
|  | Xhigh =xhigh\*-1 |
|  | Negative=true |
|  | W= xhigh/N |
| 2 | Para i=0, mientras i<=20, de i++ |
|  | X2=pow(xlow,2)/2 |
|  | Ex=exp(-x2) |
|  | Fx=ex/fija |
| 3 | If(i==0 || i==20) |
|  | Term=fx\*w/3 |
| 4 | Else If(imod2 ==0) |
|  | Term=2\*fx\*w/3 |
| 5 | else |
|  | Term=4\*fx\*w/3 |
|  | Sumterm+=term |
|  | Xlow+=w; |
| 6 | If(negative) |
|  | Sumterm-=0.5 |
| 7 | Else |
|  | Sumterm+=0.5 |
| 8 | Return sumterm |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | SECNORMAL |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | Using namespace std; | | |
|  | | Struc nodo inicio | | |
|  | | Double tablavaloresnor | | |
|  | | Double sumterm | | |
|  | | Double rglsmps | | |
|  | | Int segmentos | | |
|  | | double pk | | |
|  | | Double contsegment | | |
|  | | Double pabove | | |
|  | | Double pbelow | | |
|  | | Double xbelow | | |
|  | | Double xabove | | |
|  | | Double busqueda | | |
|  | | Double contador | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
|  | Segmentos=CALRANGOS() |
|  | Rglsmps=-4 |
|  | Sumterm=0 |
|  | Contsegment=0 |
|  | valregla=0 |
|  | Inicio =raíz; |
|  | Búsqueda = numeroy de inicio |
|  | Tablavaloresnor en posición 1,0 = -100000 |
|  | Para i=0, mientras i<segmentos, de i++ |
|  | Contsegment+=tablavaloresnor en posición 3, i |
|  | pk=1/contsegment; |
|  | Mientras pabove< pk |
|  | Pbelow=pabove |
|  | Xbelow=xabove |
|  | pabove=REGLASMPNOR(rglsmps) |
|  | Xabove=rglsmps |
|  | Rglsmps-=.1 |
|  | Tablavaloresnor en posición 2,i= xbelow+((pk-pbelow)/(pabove- pbelow))\*(xabove-xbelow) |
|  | Tablavaloresnor en posición 1,i+1= xbelow+((pk-pbelow)/(pabove- pbelow))\*(xabove-xbelow) |
|  | Contador=0 |
|  | Mientras busqueda<tablavalores en posicion 2,i |
|  | Contador++ |
|  | Inicio = sig que esta en inicio |
|  | Búsqueda= numeroy de inicio |
|  | Tablavaloresnor en posición 4,i=contador |
|  | Tablavaloresnor en posición 5,i=pow(tablavaloresnor 3,i – tablavaloresnor 4,i , 2) |
|  | Tablavaloresnor en posición 6,i=tablavaloresnor 5,i / tablavaloresnor 3,i |
|  | Sumaterm+=tablavaloresnor 6,i |
|  | Return sumterm; |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | REGLASPMx2 |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <math.h>  Using namespace std; | | |
|  | | Double xlow | | |
|  | | Double xhigh | | |
|  | | Double N | | |
|  | | Double n | | |
|  | | Double W | | |
|  | | Double tfija | | |
|  | | Double fija | | |
|  | | Double xan | | |
|  | | Double exa | | |
|  | | Double fx | | |
|  | | Double term | | |
|  | | Double sumter | | |
|  | | Bool negative | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
|  | Negative=false |
|  | n=9 |
|  | Tfija=1 |
|  | Para i=1, mientras i<n,i+=2 |
|  | Tfija=tfija\*i/2 |
|  | Tfija=tfija\*sqrt(PI) |
|  | Fija=pow(2,n/2) |
|  | N=20 |
| 1 | If(xhigh <0) |
|  | Xhigh =xhigh\*-1 |
|  | Negative=true |
|  | W= xhigh/20 |
| 2 | Para i=0, mientras i<=20, de i++ |
|  | Xan=pow(xlow,(n/2)-1) |
|  | Exa=exp(-xlow/2) |
|  | Fx=(xan\*exa)/(fija\*tfija) |
| 3 | If(i==0 || i==20) |
|  | Term=fx\*w/3 |
| 4 | Else If(imod2 ==0) |
|  | Term=2\*fx\*w/3 |
| 5 | Else |
|  | Term=4\*fx\*w/3 |
|  | Sumterm+=term |
|  | Xlow+=w; |
| 6 | If(negative) |
|  | Sumterm-=0.5 |
| 7 | Else |
|  | Sumterm+=0.5 |
| 8 | Return sumterm |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | PROBABILIDAD |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <math.h>  Using namespace std; | | |
|  | | Double prob | | |
|  | |  | | |
|  | |  | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
|  | Prob=1-prob |
|  | Return prob |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | NORMALIZACION |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <math.h>  Using namespace std; | | |
|  | | Double xavg | | |
|  | | Double varianza | | |
|  | | Double dest | | |
|  | | Double Q | | |
|  | | Double simx2 | | |
|  | | Double p | | |
|  | | Nodo inicio | | |
|  | | Int i | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
|  | Xavg=AVG() |
|  | Varianza=VARIANZA(xavg) |
|  | Varianza=sqrt(varianza) |
|  | Para i=0,mientras i<n,de i++ |
|  | Numeroy de incio=numerox de inicio-xavg/varianza |
|  | Inicio=sig que esta en inicio |
|  | Q=SECNORMAL() |
|  | P=REGLASPMx2(q) |
|  | P=PROBABILIDAD(p) |
|  | Return p |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | main |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <stdio.h>  Using namespace std; | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Freopen() |
| 2 | PEDIRDATOS() |
| 3 | BURBUJALTST() |
|  | Printf(NORMALIZACION()) |
|  | Reurn 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | N/A | | Function | BURBUJALTST |
| **INCLUDES:**  **TYPE DEFINITIONS:**  **Declaration:** | | #include <stdio.h>  Using namespace std; | | |
|  | | Struct nodo \*inicio | | |
|  | | Struct nodo \*siguiente | | |
|  | | Double datox | | |
|  | | Long long i | | |
|  | | Long long y | | |
|  | | Bool ambas=false | | |
|  | | Double datoy | | |

|  |  |
| --- | --- |
| **Reference:** |  |

|  |  |
| --- | --- |
| Logic reference numbers | Program logic, in pseudocode |
| 1 | Inicialiar i=0 |
|  | Inicializar y=0 |
|  | Inicio es igual a raíz |
| 2 | Mientras i<n |
|  | Inicializar y=i+1 |
| 3 | Auxiliar es igual a lo que apunta incio en siguiente |
|  | Mientras y<n |
| 4 | Si ambas es falsa |
|  | En caso que si |
| 5 | Si datox de incio es mayor a datox de Auxiliar |
|  | En caso que si |
|  | Datox es igual a lo que hay en datox de Auxiliar |
|  | Datoy es igual a lo que hay en datoy de Auxiliar |
|  | Datox de Auxiliar es igual a lo que hay en datox de inicio |
|  | Datoy de Auxiliar es igual a lo que hay en datoy de inicio |
|  | Datox de inicio es igual a datox |
|  | Datoy de incio es igual a datoy |
|  | En caso de que no |
|  | Si datoy de incio es mayor a datoy de siguiente |
|  | En caso que si |
|  | Datox es igual a lo que hay en datox de Auxiliar |
|  | Datoy es igual a lo que hay en datoy de Auxiliar |
|  | Datox de Auxiliar es igual a lo que hay en datox de inicio |
|  | Datoy de Auxiliar es igual a lo que hay en datoy de inicio |
|  | Datox de inicio es igual a datox |
|  | Datoy de incio es igual a datoy |
|  | Y++ |
|  | Auxiliar es igual a lo que apunta Auxiliar en siguiente |
|  | I++ |
|  | Inicio es igual a lo que apunta incio en siguiente |
|  | Inicialiar i=0 |
|  | Inicio es igual a raíz |
| 6 | Mientras i<n |
|  | Imprime el valor de x & y |