|  |
| --- |
| College LaSalle |
| Project – Student Management Technical Manual |
|  |
| Presented to: Mihai Maftei |

|  |
| --- |
| Salomon Bulambo Baraka  7/25/2022  Version 1.1 |

**(3 points)**

Start by adding a short description of your project, and the languages (technologies) used

**Languages/tools/editors**:

1. Microsoft Visual Studio Community 2022 (64-bit) Version 17.1.4
2. Code::Blocks (64-bit) Version 20.03-r11983

**Presenting the algorithm(s)**:  **(4 points)**

BEGIN

declare function print\_welcome()

WRITE "Wellcome to Project II"

WRITE "420 - AP1 - AS - PROJECT 2";

WRITE << "weights conversions Application";

WRITE << "\*---\*---\*---\*---\*---\*---\*---\*---\*";

ENDFUNCTION

declare function print\_main\_list()

WRITE "Main Menu"

WRITE "1- "Create and add a student"

WRITE "2- "Search for a student"

WRITE "3- "Modify the information of the student"

WRITE "4- "Delete the information of the student"

WRITE "5- "Display the student list"

WRITE "6- "Exit the application"

ENDFUNCTION

declare function void exit()

call function print\_welcome()

char ch

WRITE "Do you want to logout...!! Enter (Y, N)"

IF (set ch = Y | ch = y) THEN

call function print\_welcome();

WRITE "Thanks for using"

WRITE "Student Management Application"

ENDIF

ELSE

call function main()

ENDELSE

ENDFUNCTION

declare function error\_option()

WRITE "Incorrect value...!"

Return to main()

ENDFUNCTION

declare function void add()

call function print\_welcome

int counter

set counter to box

REPEAT

WRITE "Student No.", counter + 1

WRITE "ID:"

READ p[counter].ID

WRITE "First Name:"

READ p[counter].per.first\_name

WRITE "Last Name:"

READ p[counter].per.last\_name

WRITE "Age:"

READ p[counter].per.age

WRITE "Number and Street Name:"

READ p[counter].addr.nbe\_str\_name, 30

WRITE "City Name:"

READ p[counter].addr.city\_name, 20

WRITE "Province/State:"

READ p[counter].addr.province, 15

WRITE "Postal Codee:"

READ p[counter].addr.po\_code, 15

WRITE "Phone number:"

READ p[counter].phone

WRITE "Email Address:"

READ p[counter].email

UNTIL counter <= box;

ENDLOOP

increment to box

ENDFUNCTION

declare function void search()

call function print\_welcome

set lable to m

IF (set box != 0) THEN

int id

char ans

WRITE "Enter The ID:"

READ id

REPEAT

int counter

IF (set p[counter].ID = id) THEN

WRITE "ID:", p[counter].ID

WRITE "First Name:", p[counter].per.first\_name

WRITE "Last Name:" << p[counter].per.last\_name

WRITE "Age: " << p[counter].per.age

WRITE "Address:" p[counter].addr.nbe\_str\_name, ", ", p[counter].addr.city\_name, ", ", p[counter].addr.Province, ", ", p[counter].addr.po\_code

WRITE "Phone No.:", p[counter].phone

WRITE "Email Address:" << p[counter].email

WRITE "Serach Another Student Press (Y, N)"

READ ans

IF (set ans = Y | ans =y) THEN

set goto lable m

ENDIF

ELSE

call main()

ENDELSE

set stop

ENDIF

ELSEIF (set counter = box - 1) THEN

WRITE "No record found"

WRITE "Return to Main Menu"

call main()

ENDELSEIF

UNTIL set counter >= box

ENDLOOP

ELSE

WRITE "No data is entered"

WRITE "Return to Main Menu ";

cal function main()

ENDELSE

ENDFUNCTION

declare function void modify()

call function print\_welcome

IF (set box != 0) THEN

set lable to d

call function print\_welcome

int id, ch

char ans

WRITE "Modify the information of the student"

WRITE "1- "Modify First Name"

WRITE "2- "Modify Last Name"

WRITE "3- "Modify Address"

WRITE "4- "Modify Phone No."

WRITE "5- "Modify Email Address"

WRITE "6- "Return to Main Menu"

WRITE "Enter your option"

READ ch

call function print\_welcome

CASE ch OF

'1': WRITE ""Enter Student ID to be Modify:"

READ id

REPEAT

int counter

set counter to ZERO

IF (set p[counter].ID = id) THEN

WRITE "First Name:", p[counter].per.first\_name

WRITE "New First Name:"

READ p[counter].per.first\_name

WRITE "Modified successfully...!"

WRITE "Modify Another Student Press (Y, N):"

READ ans

IF (set ans = Y | ans =y) THEN

set goto lable d

ENDIF

ELSE

call main()

ENDELSE

set stop

ENDIF

ELSEIF (set counter = box - 1) THEN

WRITE "No record found"

WRITE "Return to Menu"

goto lable d

ENDELSEIF

UNTIL set counter >= box

'2': WRITE "Enter Student ID to be Modify:"

READ id

REPEAT

int counter

set counter to ZERO

IF (set p[counter].ID = id) THEN

WRITE "Last Name:", p[counter].per.last\_name

WRITE "New Last Name:"

READ p[counter].per.last\_name

WRITE "Modified successfully...!"

WRITE "Modify Another Student Press (Y, N):"

READ ans

IF (set ans = Y | ans =y) THEN

set goto lable d

ENDIF

ELSE

call main()

ENDELSE

set stop

ENDIF

ELSEIF (set counter = box - 1) THEN

WRITE "No record found"

WRITE "Return to Menu"

set goto lable d

ENDELSEIF

UNTIL set counter >= box

'3': WRITE ""Enter Student ID to be Modify:"

READ id

REPEAT

int counter

set counter to ZERO

IF (set p[counter].ID = id) THEN

WRITE "The Address:"

WRITE "Number and street name:", p[counter].addr.nbe\_str\_name

WRITE "City name:", p[counter].addr.city\_name

WRITE "Province/State:", p[counter].addr.Province

WRITE "Postal code:", p[counter].addr.po\_code

WRITE "New Number and street name:"

READ p[counter].addr.nbe\_str\_name, 30

WRITE "New City name:"

READ p[counter].addr.city\_name, 20

WRITE "New Province/State:"

READ p[counter].addr.Province, 15

WRITE "New Postal code:"

READ p[counter].addr.po\_code, 7

WRITE "Modified successfully...!"

WRITE "Modify Another Student Press (Y, N):"

READ ans

IF (set ans = Y | ans =y) THEN

set goto lable d

ENDIF

ELSE

call main()

ENDELSE

set stop

ENDIF

ELSEIF (set counter = box - 1) THEN

WRITE "No record found"

WRITE "Return to Menu"

set goto lable d

ENDELSEIF

UNTIL set counter >= box

'4': WRITE ""Enter Student ID to be Modify:"

READ id

REPEAT

int counter

set counter to ZERO

IF (set p[counter].ID = id) THEN

WRITE "Phone No.:", p[counter].phone

WRITE "New Phone No:"

READ p[counter].phone

WRITE "Modified successfully...!"

WRITE "Modify Another Student Press (Y, N):"

READ ans

IF (set ans = Y | ans =y) THEN

set goto lable d

ENDIF

ELSE

call main()

ENDELSE

set stop

ENDIF

ELSEIF (set counter = box - 1) THEN

WRITE "No record found"

WRITE "Return to Menu"

set goto lable d

ENDELSEIF

UNTIL set counter >= box

'5': WRITE ""Enter Student ID to be Modify:"

READ id

REPEAT

int counter

set counter to ZERO

IF (set p[counter].ID = id) THEN

WRITE "Email Address:", p[counter].email

WRITE "New Email Address:"

READ p[counter].email

WRITE "Modified successfully...!"

WRITE "Modify Another Student Press (Y, N):"

READ ans

IF (set ans = Y | ans =y) THEN

set goto lable d

ENDIF

ELSE

call main()

ENDELSE

set stop

ENDIF

ELSEIF (set counter = box - 1) THEN

WRITE "No record found"

WRITE "Return to Menu"

set goto lable d

ENDELSEIF

UNTIL set counter >= box

'6': call function main()

OTHERS: call function error\_option()

ENDCASE

ENDIF

ELSE

WRITE "No data is entered"

WRITE "Return to Main Menu ";

cal function main()

ENDELSE

ENDFUNCTION

declare function void del()

call function print\_welcome

IF (set box != 0) THEN

set lable to two

call function print\_welcome

int id, ch

char ans

WRITE "Delete the information of the studen"

WRITE "1- "Delete Specific Student"

WRITE "2- "Delete All Students"

WRITE "3- "Return to Main Menu"

WRITE "Enter your option"

READ ch

call function print\_welcome

CASE ch OF

'1': WRITE "Enter student ID to be delete:"

READ id

REPEAT

IF(set p[counter].ID = id) THEN

p[counter].ID = p[counter + 1].ID;

p[counter].per.first\_name = p[counter + 1].per.first\_name;

p[counter].per.last\_name = p[counter + 1].per.last\_name;

p[counter].per.age = p[counter + 1].per.age;

p[counter].addr.nbe\_str\_name = p[counter + 1].addr.nbe\_str\_name;

p[counter].addr.city\_name = p[counter + 1].addr.city\_name;

p[counter].addr.po\_code = p[counter + 1].addr.po\_code;

p[counter].addr.Province = p[counter + 1].addr.Province;

p[counter].phone = p[counter + 1].phone;

p[counter].email = p[counter + 1].email;

WRITE "Do you want to delete student No.", id, "? Press (Y, N):"

READ ans

IF (set ans = Y | ans =y) THEN

to box

WRITE "Deleted successfully...!"

WRITE "Delete Another Student Record Press (Y, N):"

READ ans

IF (set ans = Y | ans =y) THEN

set goto lable two

ENDIF

ELSE

call main()

ENDELSE

set stop

ENDIF

ELSE

set goto lable two

ENDIF

ELSEIF (set counter = box - 1) THEN

WRITE "No record found"

WRITE "Return to Menu"

set goto lable two

ENDELSEIF

UNTIL set counter >= box

'2': WRITE "Do you want to delete all Students? Press (Y, N):"

READ ans

IF (set ans = Y | ans =y) THEN

set box to ZERO

WRITE "Deleted successfully...!"

WRITE "Return to Menu"

set goto lable two

ENDIF

ELSE

set goto lable two

ENDELSE

'3': call function main()

OTHERS: call function error\_option()

ENDIF

ELSE

WRITE "No data is entered"

WRITE "Return to Main Menu ";

cal function main()

ENDELSE

ENDFUNCTION

declare function void show()

call function print\_welcome

WRITE "General List of students"

IF (set box != 0) THEN

int temp

set temp to ZERO

string ty

char ans

int counter1

set counter1 to ZERO

REPEAT

int counter2

set counter1 & counter2 to ZERO

REPEAT

IF (set p[counter1].ID < p[counter2].ID) THEN

temp = p[counter1].ID;

p[counter1].ID = p[j].ID;

p[j].ID = temp;

ty = p[counter1].per.first\_name;

p[counter1].per.first\_name = p[counter2].per.first\_name;

p[counter2].per.first\_name = ty;

ty = p[counter1].per.last\_name;

p[counter1].per.last\_name = p[counter2].per.last\_name;

p[counter2].per.last\_name = ty;

temp = p[counter1].per.age;

p[counter1].per.age = p[counter2].per.age;

p[counter2].per.age = temp;

ty == p[counter1].addr.nbe\_str\_name;

p[counter1].addr.nbe\_str\_name == p[counter2].addr.nbe\_str\_name;

p[counter2].addr.nbe\_str\_name == ty;

ty == p[counter1].addr.city\_name;

p[counter1].addr.city\_name == p[counter2].addr.city\_name;

p[counter2].addr.city\_name == ty;

ty == p[counter1].addr.Province;

p[counter1].addr.Province == p[counter2].addr.Province;

p[counter2].addr.Province == ty;

ty == p[counter1].addr.po\_code;

p[counter1].addr.po\_code == p[counter2].addr.po\_code;

p[counter2].addr.po\_code == ty;

ty = p[counter1].phone;

p[counter1].phone = p[counter2].phone;

p[counter2].phone = ty;

ty = p[counter1].email;

p[counter1].email = p[counter2].email;

p[counter2].email = ty;

ENDIF

UNTIL set counter2 >= box

ENDLOOP2

UNTIL set counter1 >= box

ENDLOOP1

int counter

set counter to ZERO

REPEAT

WRITE "Data of student No.", i + 1

WRITE "Student ID:", p[counter].ID

WRITE "Student First Name:", p[counter].per.first\_name

WRITE "Student Last Name:", p[counter].per.last\_name

WRITE "Student Age:", p[counter].per.age

WRITE "Student Address:", p[counter].addr.nbe\_str\_name, ", ", p[counter].addr.city\_name, ", ", p[counter].addr.Province, ", ", p[counter].addr.po\_code

WRITE "Student Phone No.:", p[counter].phone

WRITE "Student Email Address:"p[counter].email

UNTIL set counter >= box

ENDLOOP

WRITE "Return to Main Menu Press (Y):"

READ ans

WHILE (set to TRUE)

IF (set ans = Y | ans = y) THEN

call function main()

ENDIF

ELSE

call function main()

ENDELSE

set stop

ENDWHILE

ENDIF

ELSE

WRITE "No data is entered"

WRITE "Return to Main Menu ";

cal function main()

ENDELSE

ENDFUNCTION

declare struct person{

string first\_name, last\_name;

int age;

}

declare struct person per

declare struct address{

declare array char nbe\_str\_name[30];

declare array char city\_name[20];

declare array char Province[15];

declare array char po\_code[7];

}

declare struct address addr

declare struct student\_Management{

int id

sub struct per

sub struct addr

string phone, email

}

declare array struct student\_Management p[]

set p[] to 200

int box

set box to zero;

in option

char ans

call function print\_welcome()

call function print\_main\_list

WRITE "Enter your option"

READ option

CASE option OF

'1': REPEAT

call function add()

WRITE "Add Another Student Record Press (Y, N):"

READ ans

UNTIL (ans != Y | ans !=y)

Return to main()

'2': call function search()

'3': call function modify()

'4': call function del()

'5': call function show()

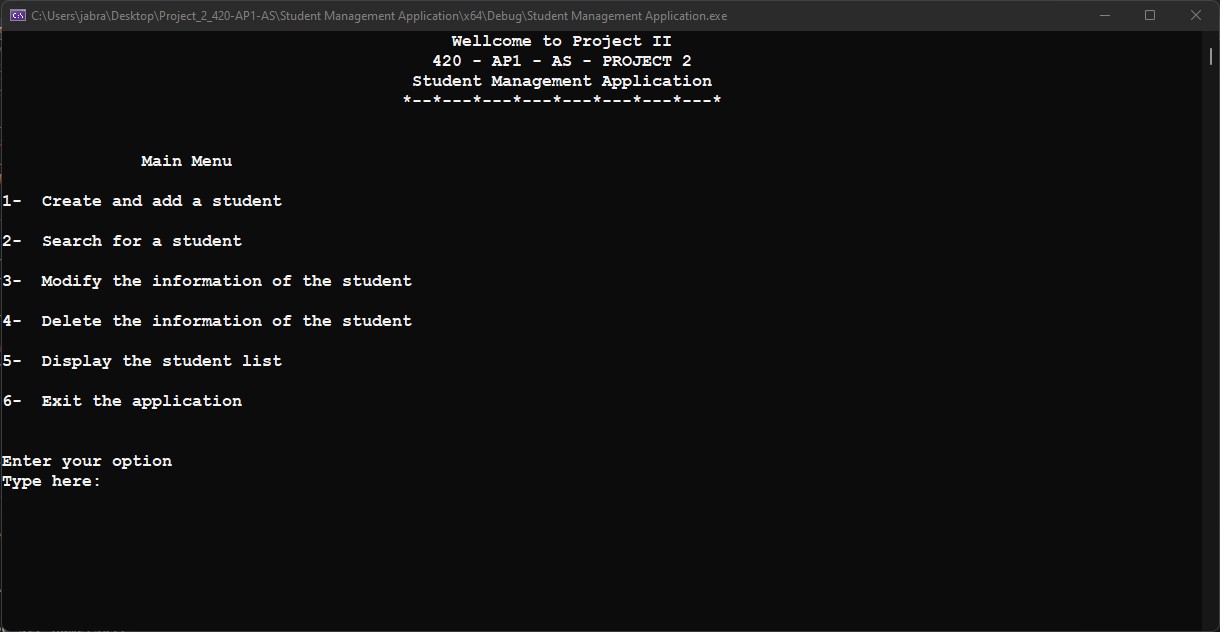
'6': call function exit()

'7': call function error\_option()

END

**Present the functions** (prototypes) that you create or did use in the project. **: (4 points)**

|  |  |
| --- | --- |
| **Function Name (prototype)** | **Description** |
| 1. void print\_welcome(); | To print of welcome text. |
| 1. void print\_main\_list() | To print of Main menu list. |
| 1. void exit(); | To exit the application with confirmation. |
| 1. void error\_option(); | To print error message when user type input wrong especially in main menu. |
| 1. void add(); | To ask user Enter a weight in Lbs and compute it up to 15 or till the user enters 0 or a negative value to stop. |
| 1. void search(); | To ask user Enter a weight in Kg and compute it up to 15 or till the user enters 0 or a negative value. |
| 1. void modify(); | To display a weight in Kg after computed it |
| 1. void modify(); | To display a weight in Lbs after computed it |
| 1. void del(); | To return Max of weights and display it |
| 1. void show(); | To return Min of weights and display it |
| 1. Switch {case 1 – 3 and default} | To return and display any function which choosed by user |
| 1. struct person {}; | To represent a record of object person |
| 1. struct address | To represent a record of object address |
| 1. struct student\_Management | To represent a record of others struct |

**Presenting the outputs (print screens) that you obtain: (4 points)**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Implement the functional test plan:**

* 1. Proper functioning of the program:

|  |
| --- |
| **Programming condition for a proper functioning of the program** |
| I use 2 string / 1 int -> struct person |
| I use 4 char as array different size -> struct address |
| I use 1 int / 2 string / 2 sub struct -> struct student\_Management |
| Global array size[200] to store all inputs elements |
| Global double variable, copy elements from array. |
| 1 int / 1 char – for the main menu |
| 1 do/while loop – main function – switch --->> case 1 |
| Function add(); – main function – switch --->> case 1 |
| Function search(); – main function – switch --->> case 2 |
| Function modify(); – main function – switch --->> case 3 |
| Function delete(); – main function – switch --->> case 4 |
| Function show(); – main function – switch --->> case 5 |
| Function exit(); – main function – switch --->> case 56 |
| Function error\_option(); – main function – switch --->> case |
| 1 char / 1 if-else / 1 for in body exit(); Function |
| 1 if-else / 2 for loop / 1 while loop in body error\_option(); Function |
| 6 while loop / 1 for loop in body add(); Function |
| 1 int/1char/3 for loop/ 3 if-else in body search(); Function |
| I use 2 int & char for the main list in body modify(); Function |
| 2 for loop/3 if / 2 else – case-1 – in body modify(); Function |
| 2 for loop/3 if / 2 else – case-2 – in body modify(); Function |
| 2 for loop/3 if / 2 else/4 while loop – case-3 – in body modify(); Function |
| 2 for loop/3 if / 2 else – case-4 – in body modify(); Function |
| 2 for loop/3 if / 2 else – case-5 – in body modify(); Function |
| 3 for loop/1 if / 2 else/1 while loop – default – in body modify(); Function |
| I use 2 int & char for the main list in body delete(); Function |
| 2 for loop/4 if / 3 else – case-1 – in body delete (); Function |
| 1 for loop/1 if / 1 else – case-2 – in body delete (); Function |
| 3 for loop/1 if / 2 else/1 while loop – default – in body delete(); Function |
| 4 for loop/2 if / 2 else/1 while loop –in body show(); Function |
|  |

* 1. Clear record of information concerning tests and their results:

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **What testing?** | **Testing Sample (input – output)** | **Result** |
| 2022-04-16 | print\_welcome(); | Display Wellcome to Project II……more | v |
| 2022-04-16 | print\_main\_list(); | Display Main Menu | v |
| 2022-04-16 | Test choose option | cout Enter your option / cin option | v |
| 2022-04-16 | Test case 3 exit(); | Confirmation text yes or no | x |
| 2022-04-16 | Test case 3 exit(); | Confirmation text yes or no with timer | x |
| 2022-04-16 | Test case 3 exit(); | Confirmation text yes or no with timer | v |
| 2022-04-16 | Test error\_option(); | Incorrect value & Return to Main Menu | x |
| 2022-04-16 | Test error\_option(); | Incorrect value & Return to Main Menu  With timer | v |
| 2022-04-16 | Test function add(); | cin >> p[i].ID; …. | x |
| 2022-04-16 | Test function add(); | cin >> p[i].ID;….. | v |
| 2022-04-16 | Test function add(); | while (!cin.good()) {cerr << "Incorrect value...! Please try again\n"; | x |
| 2022-04-16 | Test function add(); | while (!cin.good()) {cerr << "Incorrect value...! Please try again\n";  box++; | v |
| 2022-04-16 | Test function search(); | if (p[i].ID == id) {cout << "ID: " << p[i].ID << endl; ….. | x |
| 2022-04-16 | Test function search(); | if (p[i].ID == id) {cout << "ID: " << p[i].ID << endl; ….. | v |
| 2022-04-16 | Test function modify(); | if (p[i].ID == id) {cout << "First Name: " << p[i].per.first\_name << "\n\n"; … | x |
| 2022-04-16 | Test function modify(); | if (p[i].ID == id) {cout << "First Name: " << p[i].per.first\_name << "\n\n"; … | v |
| 2022-04-16 | Test function modify(); case-1 | else if (i == box - 1) {cout << "No record found" << endl; cout << "\nReturn to Menu "; | v |
| 2022-04-16 | Test function delete(); case-1 | cout << "1- Delete Specific Student\n\n"; | v |
| 2022-04-16 | Test function delete (); case-1 | if (p[i].ID == id) {p[i].ID = p[i + 1].ID;….. | x |
| 2022-04-16 | Test function delete (); case-1 | if (p[i].ID == id) {p[i].ID = p[i + 1].ID;….. | v |
| 2022-04-16 | Test function delete (); case-1 | Confirmation to delete if (ans == 'Y' || ans == 'y') {box--; | x |
| 2022-04-16 | Test function delete (); case-1 | Confirmation to delete if (ans == 'Y' || ans == 'y') {box--; | x |
| 2022-04-16 | Test function delete (); case-1 | Confirmation to delete if (ans == 'Y' || ans == 'y') {box--; | x |
| 2022-04-16 | Test function delete (); case-2 | Confirmation to delete all if (ans == 'Y' || ans == 'y') {box = 0; | v |
| 2022-04-16 | Test function show(); | if (p[i].ID < p[j].ID) {temp = p[i].ID; p[i].ID = p[j].ID; p[j].ID = temp; | x |
| 2022-04-16 | Test function show(); | ty = p[i].per.first\_name; p[i].per.first\_name = p[j].per.first\_name; p[j].per.first\_name = ty; | x |
| 2022-04-16 | Test function show(); | if (p[i].ID < p[j].ID) {temp = p[i].ID; p[i].ID = p[j].ID; p[j].ID = temp; | v |
| 2022-04-16 | Test function show(); | ty = p[i].per.first\_name; p[i].per.first\_name = p[j].per.first\_name; p[j].per.first\_name = ty; | v |
| 2022-04-16 | Test function show(); | cout << "\nReturn to Main Menu Press (Y): "; | V |
| 2022-04-16 | Test function show(); | while (true){if (ans == 'Y' || ans == 'y') {main(); }else {main();}break; | v |