QUESTION 1:

#include<iostream>

#include<conio.h>

#include<stdio.h>

using namespace std;

class Bank

{

public:

char name[20];

char account\_type[20];

int account\_number;

int balance;

void assign()

{

cout << "Enter the name of account owner:";

cin >> name;

cout << "Enter type of Account :";

cin >> account\_type;

cout << "Enter account number:";

cin >> account\_number;

cout << "Enter balance to deposit:";

cin >> balance;

}

void deposit()

{

int bal;

cout << "\nEnter the amout to deposit:";

cin >> bal;

balance += bal;

cout << "\nAmount deposited successfuly\nYour New Balance is :" << balance;

}

void balancecheck()

{

int bal;

cout << "\nYour balance :" << balance << "\nEnter amount to withdraw:";

cin >> bal;

if (bal <= balance)

{

balance -= bal;

cout << "\nRemaining Balance:" << balance;

}

else

{

exit(0);

}

}

void display()

{

cout << "\nName :";

cout << name;

cout << "\nBalance :" << balance;

cout << "\n Account number :" << account\_number;

cout << "\n Account Type : " << account\_type;

}

};

int main()

{

int i;

Bank n;

n.assign();

cout << "\n1. Your Information\n2. Deposit\n3. Withdraw\n\nEnter your choice : \n";

cin >> i;

if (i == 1)

{

n.display();

}

else if (i == 2)

{

n.deposit();

}

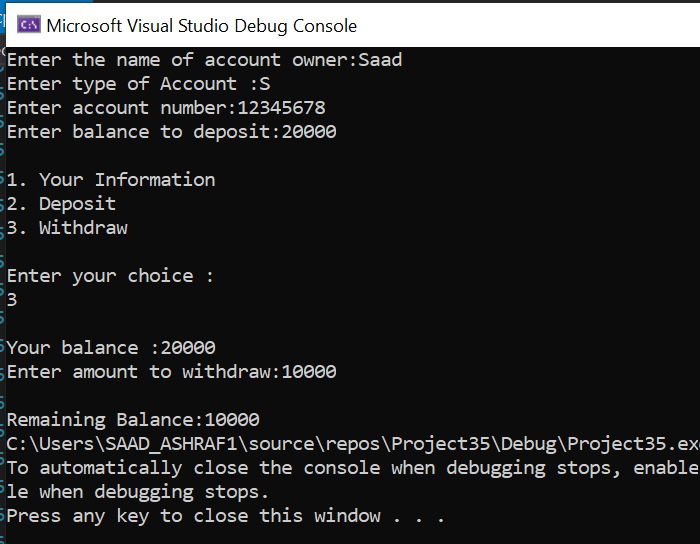
else if (i == 3)

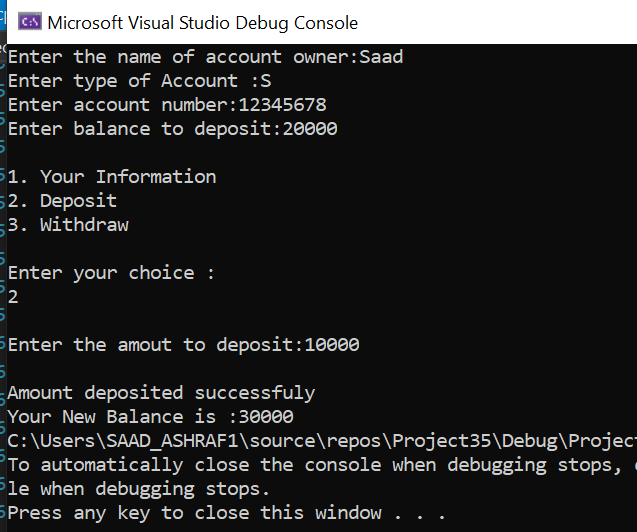
{

n.balancecheck();

}

}





QUSETION NO 2:

In C++ programming, this is a keyword that refers to the current instance of the class. There can be 3 main usage of this keyword in C++. It can be used to pass current object as a parameter to another method. It can be used to refer current class instance variable.

#include<iostream>

using namespace std;

class abc

{

private:

int x;

public:

void output() {

cout << "Printing value of : a = " << x << endl;

}

void initalize(int x)

{

this->x = x;

}

};

int main()

{

abc n;

int a = 30;

n.initalize(a);

n.output();

return 0;

}

It is also used to return reference of the calling object.

abc& abc::func ()

{

**return** \***this**;

}

QUESTION NO 3:

#include <iostream>

#include <string>

#include <iomanip>

#include <vector>

using namespace std;

class HotDogStand

{

private:

int standsID;

int hotDogsSold;

static int totalHotDogsSold;

public:

HotDogStand()

{

standsID = 0;

hotDogsSold = 0;

}

HotDogStand(int newStandID, int newHotDogsSold)

{

standsID = newStandID;

hotDogsSold = newHotDogsSold;

HotDogStand::totalHotDogsSold += newHotDogsSold;

}

void justSold()

{

hotDogsSold++;

HotDogStand::totalHotDogsSold++;

}

int getHotDogsSold()

{

return hotDogsSold;

}

static int getTotalHotDogsSold()

{

return HotDogStand::totalHotDogsSold;

}

int getStandsID()

{

return standsID;

}

static void main(std::vector<std::string>& args)

{

int x, y, a, b, c, d;

cin >> x;

cin >> y;

cin >> a;

cin >> b;

cin >> c;

cin >> d;

HotDogStand\* dog1 = new HotDogStand(x, y);

HotDogStand\* dog2 = new HotDogStand(a, b);

HotDogStand\* dog3 = new HotDogStand(c, d);

dog1->justSold();

dog1->justSold();

dog1->justSold();

dog2->justSold();

dog2->justSold();

dog2->justSold();

dog2->justSold();

dog2->justSold();

dog3->justSold();

dog3->justSold();

cout << "StandsID Hot Dogs Sold" << endl;;

cout << "-----------------------" <<endl;

cout << dog1->getStandsID()<<" " << dog1->getHotDogsSold() << endl;;

cout << dog2->getStandsID()<<" " << dog2->getHotDogsSold() << endl;

cout << dog3->getStandsID()<<" " << dog3->getHotDogsSold() << endl;

cout << "The Total number of hot dogs sold by all hot dog stands: " + std::to\_string(HotDogStand::getTotalHotDogsSold()) <<endl;

}

};

int HotDogStand::totalHotDogsSold = 0;

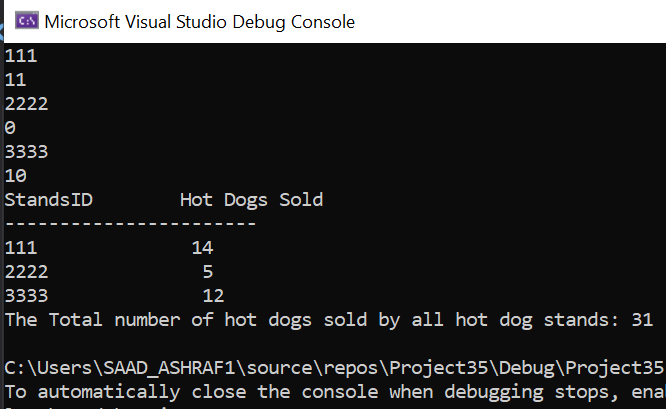
int main(int argc, char\*\* argv) {

std::vector<std::string> parameter(argv + 1, argv + argc);

HotDogStand::main(parameter);

return 0;

};



QUESTION NO 4:

#include<iostream>

using namespace std;

int MAX(int a, int b)

{

return a + b;

}

void print(int(\*C[])(int, int))

{

int i = 0;

for (i = 0; i < 4; i++)

{

cout << C[i](i + 1, i + 2);

cout << endl;

}

cout << "Print Function";

}

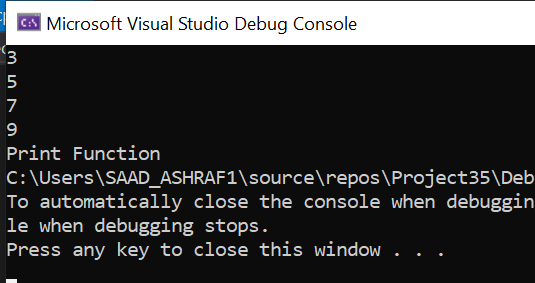
int main()

{

int (\*t[])(int, int) = { MAX, MAX, MAX, MAX };

print(t);

}



QUESTION NO 5:

#include <iostream>

using namespace std;

int series(int x) {

if ( (x != 0) || (x != 1) ) {

return x;

}

else {

return (series(x - 1) + series(x - 2));

}

}

int main() {

int x, i = 0;

cout << "Enter the number upto which series is to be calculated : ";

cin >> x;

cout << "The Fibonnaci Series is : ";

while (i < x) {

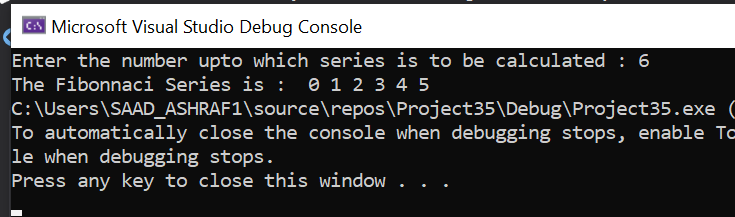
cout << " " << series(i);

i++;

}

return 0;

}



# A Graphical Representantion of the recersive code of Febonocii Series

