**DELIVERABLE FOR FINAL PROJECT**

**Introduction:**

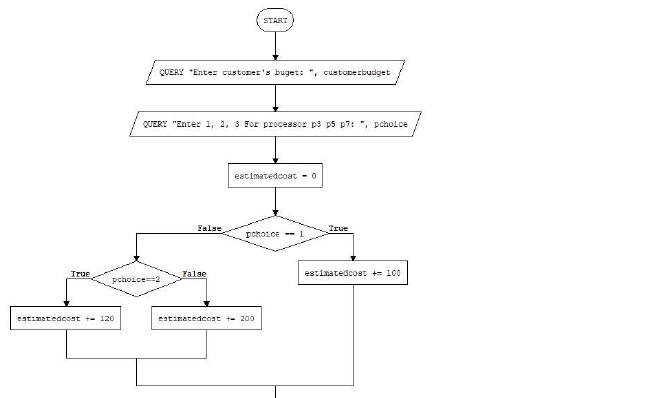
In this program we have to solve a problem of type, A computer shop will build a computer from components to meet a customer’s requirements. For each request for a computer to be built, an estimate of the cost is produced. The shop will check if the price is within the client’s budget and suggest a way to make it within the budget.

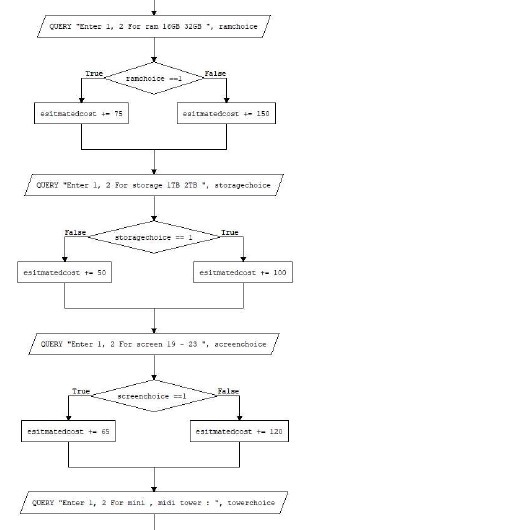
Here, first of all we’ll input budget of buyer want to buy a computer. Then we’ll show a menu to the user what user want to buy. Program will give 3 choices for processor whether p3, p5 or p7. Then there will be a choice of RAM size whether 16 GB or 32 GB. Then user will get option for storage size. There will be two options 1TB and 2TB. Then program will also offer choices for case and USB ports.

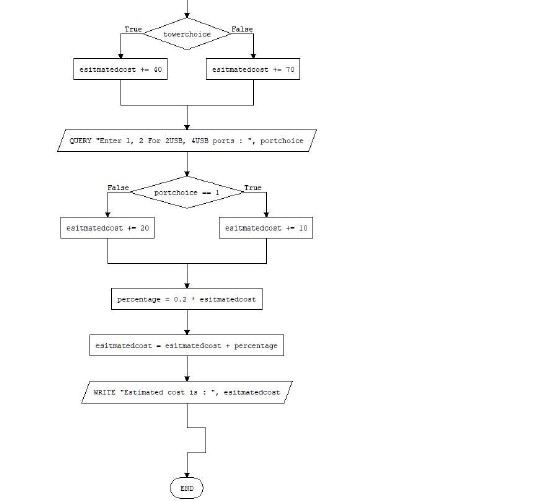
The user will make a choice for each component and an estimate is produced. Estimated cost would be return from the function. Then we need to compare it with customer budget. If estimated cost is under customer budget It will be fine. Otherwise, it’ll request the customer to reduce the cost.

**Flow Chart:**

Flow chart for a specific problem is given below.



****

****

**Code:**

#include <iostream>

using namespace std;

double purchase()

{

    cout << "Enter 1. For P3 processor" << endl;

    cout << "Enter 2. For P5 processor" << endl;

    cout << "Enter 3. For P7 processor" << endl;

    cout << "Select Processor : ";

    double estimatedcost = 0;

    int processorChoice;

    int processorPrice = 0;

    cin >> processorChoice;

    switch (processorChoice)

    {

    case 1:

        processorPrice = 100;

        break;

    case 2:

        processorPrice = 120;

        break;

    case 3:

        processorPrice = 200;

        break;

    default:

        cout << "Invalid choice!";

        break;

    }

    cout << "Enter 1. For 16 GB ram" << endl;

    cout << "Enter 2. For 32 GB ram" << endl;

    cout << "Select RAM : ";

    int ramchoice;

    int ramprice = 0;

    cin >> ramchoice;

    switch (ramchoice)

    {

    case 1:

        ramprice = 75;

        break;

    case 2:

        ramprice = 150;

        break;

    default:

        cout << "Invalid choice!";

        break;

    }

    cout << "Enter 1. For 1 TB Storage" << endl;

    cout << "Enter 2. For 2 TB Storage" << endl;

    cout << "Select Storage : ";

    int storagechoice;

    int storageprice = 0;

    cin >> storagechoice;

    switch (storagechoice)

    {

    case 1:

        storageprice = 50;

        break;

    case 2:

        storageprice = 100;

        break;

    default:

        cout << "Invalid choice!";

        break;

    }

    cout << "Enter 1. For 19 \" Screen" << endl;

    cout << "Enter 2. For 23 \" Screen" << endl;

    cout << "Select Screen : ";

    int screenchoice;

    int screenprice = 0;

    cin >> screenchoice;

    switch (screenchoice)

    {

    case 1:

        screenprice = 65;

        break;

    case 2:

        screenprice = 120;

        break;

    default:

        cout << "Invalid choice!";

        break;

    }

    cout << "Enter 1. For Mini Tower Case" << endl;

    cout << "Enter 2. For Midi Tower Case" << endl;

    cout << "Select Case : ";

    int casechoice;

    int caseprice = 0;

    cin >> casechoice;

    switch (casechoice)

    {

    case 1:

        caseprice = 40;

        break;

    case 2:

        caseprice = 70;

        break;

    default:

        cout << "Invalid choice!";

        break;

    }

    cout << "Enter 1. For 2 USB ports" << endl;

    cout << "Enter 2. For 4 USB ports" << endl;

    cout << "Select USB ports : ";

    int portchoice;

    int portprice = 0;

    cin >> portchoice;

    switch (portchoice)

    {

    case 1:

        portprice = 10;

        break;

    case 2:

        portprice = 20;

        break;

    default:

        cout << "Invalid choice!";

        break;

    }

    estimatedcost = processorPrice + ramprice + storageprice + screenprice + caseprice + portprice;

    double cost = 0.2 \* estimatedcost;

    estimatedcost = cost + estimatedcost;

    return estimatedcost;

}

bool esimateBudget(double estimate, double current)

{

    if (estimate > current)

    {

        cout << "Price is not in your Budget" << endl;

        cout << "Kindly reduce the cost" << endl;

        return false;

    }

    return true;

}

int main()

{

    double currentbudget;

    cout << "Enter your budget : ";

    cin >> currentbudget;

    double estimateCost = purchase();

    cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

    cout << "Total estimated cost : " << estimateCost << endl;

    cout << "Total Current Budget : " << currentbudget << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    while (!esimateBudget(estimateCost, currentbudget))

    {

        estimateCost = purchase();

        cout << "\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

        cout << "Total estimated cost : " << estimateCost << endl;

        cout << "Total current cost : " << currentbudget << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

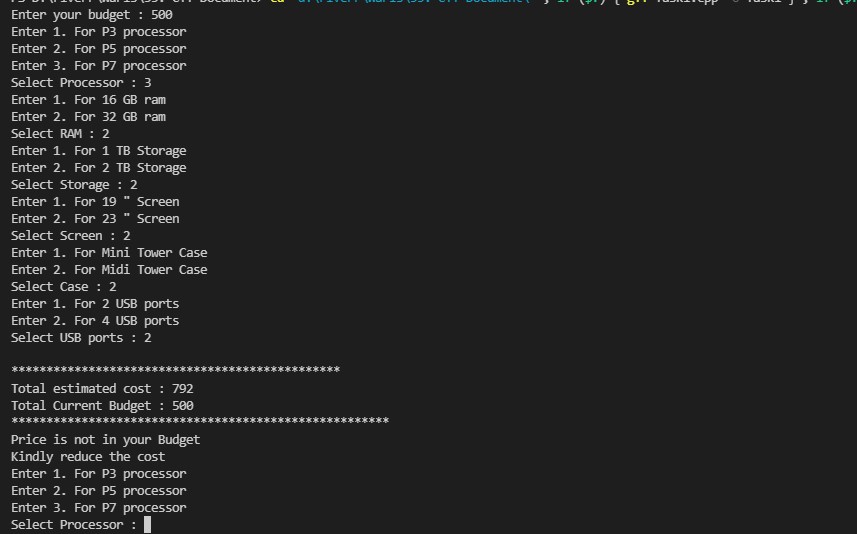
    }

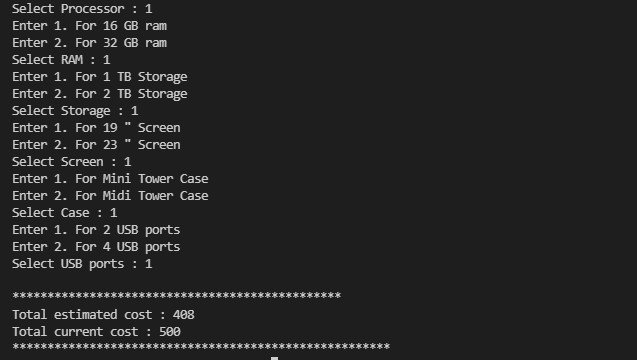
    return 0;

}

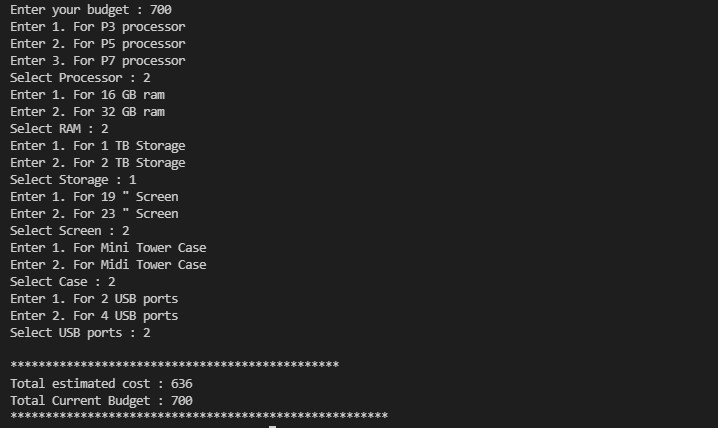
**Test Cases:**

**Run1.**

****

****

**Run2:**

****

**Discussion & Analysis:**

This program calculates the cost of building a computer using all components one by one. Customer can add component on his choice. There will be options for customer which quality or size of component one wants in computer.

The customer makes a choice for each component and an estimate is produced. The estimate must show a unique estimate number, the components chosen and the price of each component. The estimate must also show the total cost of the computer, which is calculated as the sum of the cost of the components chosen plus 20%.

In this running program, it will ask the customer for budget. It would be store in customer budget. Then it will ask for options for each component of computer.

**First** of all, it will ask for processor. Available processors are p3, p5 and p7.

If customer select p3, then 100 will be added to total computer cost.

If customer select p5, then 120 will be added to total computer cost.

If customer select p7, then 200 will be added to total computer cost.

**Secondly**, it will give options to customer for size of RAM. There will be two options for the customer. Whether of size 16 GB or size 32 GB.

If customer select 16 GB, then 75 will be added to total computer cost.

If customer select 32 GB, then 150 will be added to total computer cost.

**Thirdly**, it will give options to customer for size of Storage (ROM). There will be two options for the customer. Whether of size 1 TB or size 2 TB.

If customer select 1 TB, then 50 will be added to total computer cost.

If customer select 2 TB, then 100 will be added to total computer cost.

**Fourthly,** it will give options to customer for size Screen. There will be two options for the customer. Whether of size 19 inches or size 23 inches.

If customer select 19 inches, then 65 will be added to total computer cost.

If customer select 23 inches, then 120 will be added to total computer cost.

**Fifthly**, it will give options to customer for type of tower. There will be two options for the customer. Whether Mini Tower or Midi Tower.

If customer select Type Mini Tower, then 40 will be added to total computer cost.

If customer select Type Midi Tower, then 70 will be added to total computer cost.

**Lastly,** it will give options to customer for how many USB ports customer wants in computer. There will be two options for the customer. If user wants 2 ports customer will enter 1 otherwise 2 for 4 ports.

If customer select 2 USB ports, then 10 will be added to total computer cost.

If customer select 4 USB ports, then 20 will be added to total computer cost.

Then program will check whether total cost of computer is within range of customer. If cost is greater than customer budget, it will ask customer to reselect items within customer’s budget.

It will ask again and again until it does not meet budget.

**Conclusion & Summary:**

This program is for computer shop which help in purchasing components of computer by selecting one by one. A computer shop builds a computer from components to meet a customer’s requirements. For each request for a computer to be built, an estimate of the cost is produced. The shop will check if the price is within the client’s budget and suggest a way to make it within the budget.

The customer makes a choice for each component and an estimate is produced. Estimated cost would be return from the function. Then we need to compare it with customer budget. If estimated cost is under customer budget It will be fine. Otherwise, it’ll request the customer to reduce the cost.

**Self-Reflection:**

At our start we analyses the problem. After understanding this computer problem, we tried to make a logic to solve this problem. Then all members did separate code for problem. After discussion on solution of each member we make a one best solution. This was a way how we did our project.

The concept of problem was very good. The most interesting part of the project is that when program compare budget of customer and total cost of computer which is selected by customer. If budget does not meet it will ask customer again.

Most parts of problem were pretty logical. It takes time to get understood with problem. Where I did validation to check budget of customer and cost of computer was tricky. But we tried my best and solve again and again. Then we overcome.

Now I’ll be ready to do this type of all problem. This problem prepare myself for next time. It also teaches me how to get familiar with problem. And how to start problem step by step.