OS PHASE 1:

Group No: SY33

GROUP MEMEBRS:

53. Mayuresh Kaulwar

54. Yash Kawtikwar

57. Samyak Khadse

58. Anurag Khandare

CODE:

#include <iostream>

#include <bits/stdc++.h>

using namespace std;

int IC[2];

string Memory[100][4];

string R[4];

string IR[4];

string buff;

bool C;

ifstream read\_my\_file;

ofstream write\_my\_file;

void LOAD();

void INITIALIZE();

void STARTEXECUTION();

void MOS(int);

void READ();

void WRITE();

void TERMINATE();

void EXECUTEUSERPROGRAM();

void INITIALIZE()

{

    for (int i = 0; i < 100; i++)

    {

        for (int j = 0; j < 4; j++)

        {

            Memory[i][j] = " ";

        }

    }

    for (int i = 0; i < 4; i++)

    {

        R[i] = " ";

        IR[i] = " ";

    }

    IC[0] = 0;

    IC[1] = 0;

    C = false;

}

void LOAD()

{

    int m = 0;

    while (read\_my\_file)

    {

        getline(read\_my\_file, buff);

        string s = buff.substr(0, 4);

        if (s == "$AMJ")

        {

            INITIALIZE();

        }

        else if (s == "$DTA")

        {

            STARTEXECUTION();

        }

        else if (s == "$END")

        {

            continue;

        }

        else if (s != "$AMJ" && s != "$DTA" && s != "$END")

        {

            int j = 0;

            for (int i = 0; i < buff.size(); i++)

            {

                if (j == 4)

                {

                    j = 0;

                    m++;

                }

                if (buff[i] == 'H')

                {

                    int k = 1;

                    Memory[m][j] = buff[i];

                    for (k = 1; k <= 3; k++)

                    {

                        Memory[m][k] = ' ';

                    }

                    j = k - 1;

                }

                else

                    Memory[m][j] = buff[i];

                j++;

            }

            m = m + 1;

        }

        else

            exit(0);

    }

}

void STARTEXECUTION()

{

    IC[0] = 0;

    IC[1] = 0;

    EXECUTEUSERPROGRAM();

}

void EXECUTEUSERPROGRAM()

{

    int u1 = 0;

    while (1)

    {

        int row = IC[0] \* 10 + IC[1];

        for (int i = 0; i < 4; i++)

            IR[i] = Memory[row][i];

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

        // {

        //     cout << "IR[" << i << "] " << IR[i] << "  ";

        // }

        // cout << "\nIC:- ";

        // for (int i = 0; i < 2; i++)

        // {

        //     // cout << "IC[" << i << "] " << IC[i] << "  ";

        // }

        if (IC[1] == 9)

        {

            IC[0] = IC[0] + 1;

            IC[1] = 0;

        }

        else

            IC[1] = IC[1] + 1;

        string instruction = IR[0] + "" + IR[1];

        if (instruction == "LR")

        {

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

            {

                R[i] = Memory[stoi(IR[2]) \* 10 + stoi(IR[3])][i];

                // cout<<R[i]<<endl;

            }

        }

        else if (instruction == "SR")

        {

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

            {

                Memory[stoi(IR[2]) \* 10 + stoi(IR[3])][i] = R[i];

            }

        }

        else if (instruction == "CR")

        {

            int flag = 0;

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

            {

                if (Memory[stoi(IR[2]) \* 10 + stoi(IR[3])][i] == R[i])

                    flag++;

            }

            if (flag == 4)

            {

                cout << "\nFlag is True\n";

                C = true;

            }

            else

            {

                cout << "\nFlag is False\n";

                C = false;

            }

        }

        else if (instruction == "BT")

        {

            if (C == true)

            {

                IC[0] = stoi(IR[2]);

                IC[1] = stoi(IR[3]);

                // cout<<IC[0]<<IC[1];

            }

        }

        else if (instruction == "GD")

        {

            u1 = 1;

            MOS(u1);

        }

        else if (instruction == "PD")

        {

            u1 = 2;

            MOS(u1);

        }

        else if (instruction == "H ")

        {

            u1 = 3;

            MOS(u1);

            break;

        }

        else

            break;

    }

}

void READ()

{

    IR[3] = '0';

    getline(read\_my\_file, buff);

    // cout << buff << endl;

    int x = stoi(IR[2]) \* 10;

    int j = 0;

    // cout << x<<endl;

    for (int i = 0; i < buff.size(); i++)

    {

        if (j == 4)

        {

            j = 0;

            x++;

        }

        Memory[x][j] = buff[i];

        j++;

    }

}

void WRITE()

{

    IR[3] = '0';

    int x = stoi(IR[2]) \* 10;

    // cout << endl;

    for (int i = 0; i < 10; i++)

    {

        for (int j = 0; j < 4; j++)

        {

            // cout << Memory[x][j];

            write\_my\_file << Memory[x][j];

        }

        // cout << endl;

        x++;

    }

    write\_my\_file << endl;

}

void TERMINATE()

{

    cout << endl;

    for (int i = 0; i < 100; i++)

    {

        cout << "M[" << i << "] ";

        for (int j = 0; j < 4; j++)

        {

            cout << Memory[i][j] <<" ";

        }

        cout << endl;

    }

    // write\_my\_file << endl;

    // write\_my\_file << endl;

    LOAD();

}

void MOS(int u1)

{

    switch (u1)

    {

    case 1:

        READ();

        break;

    case 2:

        WRITE();

        break;

    case 3:

        TERMINATE();

        break;

    }

}

int main()

{

    read\_my\_file.open("input1.txt");

    write\_my\_file.open("output1.txt");

    LOAD();

    read\_my\_file.close();

    write\_my\_file.close();

}