**OS COURSE PROEJCT PHASE 1**

**SY Div:** C **Batch:** 2 **Group No.:** 10

**Members:**

Vedant Mude Rutuj Nagrale Aarya Nirgude Tejashri Nirmal

**Code:**

#include<bits/stdc++.h>

using namespace std;

void clearbuff();

void loadtoMemory(int word\_no);

void MOS();

void executeUserProgram();

void startExecution();

void init();

void LOAD();

class CPU

{

public:

char R[4];

char IR[4];

int IC[2];

bool C;

static CPU \* getObject()

{

return object;

}

private:

CPU(){};

static CPU \* object;

};

CPU\* CPU::object = new CPU;

CPU \*cpu = CPU::getObject();

char M[100][4];

char BUFF[40];

int SI=0;

fstream input;

fstream output;

void clearbuff()

{

for(auto i=0;i<40;i++)

{

BUFF[i]='-';

}

}

void init()

{

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

{

cpu->R[i]='-';

cpu->IR[i]='-';

if(i<2)

cpu->IC[i]='-';

}

cpu->C=0;

clearbuff();

for(int i=0;i<100;i++)

{

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

{

M[i][j] = '-';

}

}

}

void LOAD()

{

string line;

while(!input.eof())

{

getline(input,line);

for(auto i=0;i<line.size() && i<40;i++)

{

BUFF[i]=line[i];

}

if(BUFF[0]=='$' && BUFF[1]=='A' && BUFF[2]=='M' && BUFF[3]=='J')

{

init();

int i=0;

label:

getline(input,line);

for(auto i=0;i<line.size() && i<40;i++)

{

BUFF[i]=line[i];

}

if(!(BUFF[0]=='$' && BUFF[1]=='D' && BUFF[2]=='T' && BUFF[3]=='A'))

{

loadtoMemory(i);

i+=10;

goto label;

}

}

if(BUFF[0]=='$' && BUFF[1]=='D' && BUFF[2]=='T' && BUFF[3]=='A')

{

clearbuff();

startExecution();

}

if(BUFF[0]=='$' && BUFF[1]=='E' && BUFF[2]=='N' && BUFF[3]=='D')

{

cout<<"Memory start:"<<endl;

for(int i=0;i<100;i++)

{

cout<<" ";

if (i<10) cout<<'0';

cout<<i<<" | ";

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

{

cout<<M[i][j]<<" | ";

}

cout<<endl;

if (i%10 == 9) cout<<" -----------------"<<endl;

}

cout<<"Memory end"<<endl<<endl<<endl;

}

}

}

void loadtoMemory(int word\_no)

{

int k=0,j=0;

while(BUFF[k]!='-' && k!=40)

{

M[word\_no][j]=BUFF[k];

//to differentiate between the H instruction and 'H' string

if(BUFF[k]=='H'&& SI!=1)

{

j+=3;

}

k++,j++;

if(j==4)

{

j=0;

word\_no++;

}

}

clearbuff();

}

void startExecution()

{

cpu->IC[0]=0;

cpu->IC[1]=0;

executeUserProgram();

}

void executeUserProgram()

{

while(1)

{

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

{

cpu->IR[i]=M[cpu->IC[0]\*10+cpu->IC[1]][i];

}

cpu->IC[1]+=1;

if(cpu->IC[1]==10)

{

cpu->IC[1]=0;

cpu->IC[0]+=1;

}

if(cpu->IR[0]=='G' && cpu->IR[1]=='D')

{

SI=1;

MOS();

}

else if(cpu->IR[0]=='P' && cpu->IR[1]=='D')

{

SI=2;

MOS();

}

else if(cpu->IR[0]=='L' && cpu->IR[1]=='R')

{

int word\_no=(cpu->IR[2]-'0')\*10+(cpu->IR[3]-'0');

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

{

cpu->R[i]=M[word\_no][i];

}

}

else if(cpu->IR[0]=='S' && cpu->IR[1]=='R')

{

int word\_no=(cpu->IR[2]-'0')\*10+(cpu->IR[3]-'0');

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

{

M[word\_no][i]=cpu->R[i];

}

}

else if(cpu->IR[0]=='C' && cpu->IR[1]=='R')

{

int word\_no=(cpu->IR[2]-'0')\*10+(cpu->IR[3]-'0');

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

{

if(M[word\_no][i]==cpu->R[i])

{

cpu->C=1;

}

else

{

cpu->C=0;

break;

}

}

}

else if(cpu->IR[0]=='B' && cpu->IR[1]=='T')

{

if(cpu->C==1)

{

cpu->IC[0]=cpu->IR[2]-'0';

cpu->IC[1]=cpu->IR[3]-'0';

}

}

else if(cpu->IR[0]=='H')

{

SI=3;

MOS();

break;

}

}

}

void MOS()

{

if(SI==1)

{

cpu->IR[3]='0';

int word\_no=(cpu->IR[2]-'0')\*10;

string line;

getline(input,line);

for(auto i=0;i<line.size() && i<40;i++)

{

BUFF[i]=line[i];

}

loadtoMemory(word\_no);

SI = 0;

}

else if(SI==2)

{

cpu->IR[3]='0';

int row=(cpu->IR[2]-'0')\*10

;

for(int i=row;i<row+10;i++)

{

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

{

output<<M[i][j];

}

}

output<<endl;

SI = 0;

}

else if(SI==3)

{

SI = 0;

output<<endl<<endl;

}

}

int main()

{

input.open("input.txt",ios::in);

output.open("output.txt",ios::out);

LOAD();

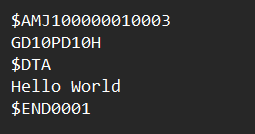
input.close();

output.close();

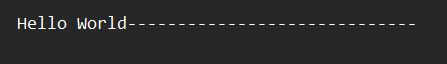
return 0;

}

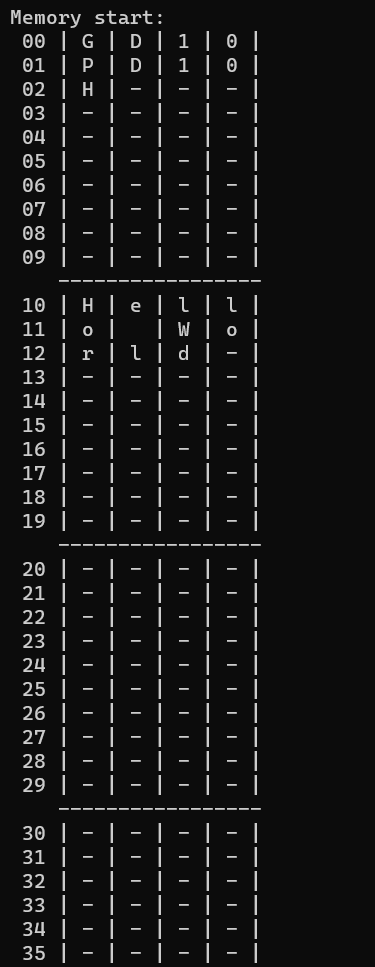
**Input File-1:**



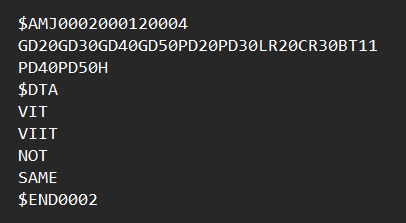
**Output File-1:**

****

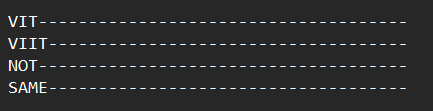
**Output Screen:** (Memory)



**Input File-2:**



**Output File-2:**



**Output Screen:** (Memory)

