```
/*************************************************************
Author: walking
Date: 2013_11_4
Description: a grid world apply state-value function
Input: nothing
Output: a state-value table
*************************************************************/
#include <iostream>
#include <stdlib.h>
#include <stdio.h>
#include <iomanip>
using namespace std;

void Bellman_eq();
void write_file(char * file_name);

double state_value[7][7]={0};           //state value table
double next_state_value[7][7]={0};      //next state value table
double reward [7][7]=                   //Reward table
{
    -1,-1,-1,-1,-1,-1,-1,
    -1, 0, 0, 0, 0, 0,-1,
    -1, 0, 0, 0, 0, 0,-1,
    -1, 0, 0, 0, 0, 0,-1,
    -1, 0, 0, 0, 0, 0,-1,
    -1, 0, 0, 0, 0, 0,-1,
    -1,-1,-1,-1,-1,-1,-1
};
int times,count=0;                      //accumulated Iteration times

int main(){
    while(cout<<"input cycle times:",cin>>times){
        system("cls");
        for(int i=0;i<times;i++,++count)Bellman_eq();//execute "times" times

        cout<<endl<<" Iteration times:"<<count<<endl<<endl;

        for(int i=1;i<6;i++){
            for(int j=1;j<6;j++)
                cout<<setprecision(1)<<fixed<<setw(6)<< state_value[i][j]<<"  ";
            cout<<endl<<endl;
        }//show state value table
        cout<<"-------------------"<<endl;
        write_file("state_value_table.csv");
    }
}

/*******************************************************************************
       Name:  Bellman_eq().
```

```
   Function:  calculate all state-value with three different V(s') condition.
 Parameters:  none.
********************************************************************************/
void Bellman_eq(){

    double up,down,left,right;
    double up_reward,down_reward,left_reward,right_reward;
    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){

            ///V(s') three condition : normal,hit wall,A.B special case
            up=state_value[i][j+1];                         //normal condition
            down=state_value[i+2][j+1];
            left=state_value[i+1][j];
            right=state_value[i+1][j+2];

            up_reward = reward[i][j+1];
            down_reward = reward[i+2][j+1];
            left_reward = reward[i+1][j];
            right_reward = reward[i+1][j+2];

            if(reward[i][j+1]==-1)      up=state_value[i+1][j+1];     //hit wall condition
            if(reward[i+2][j+1]==-1)    down=state_value[i+1][j+1];
            if(reward[i+1][j]==-1)      left=state_value[i+1][j+1];
            if(reward[i+1][j+2]==-1)    right=state_value[i+1][j+1];

            if ((i+1)==1 && (j+1)==2){                  //A[1][2],B[1][4]special case
                    up=state_value[5][2];               //any move result in [5][2],[3][4],reward 10,5.
                    down=state_value[5][2];
                    left=state_value[5][2];
                    right=state_value[5][2];
                    up_reward = 10;
                    down_reward = 10;
                    left_reward = 10;
                    right_reward = 10;
            }
            if ((i+1)==1 && (j+1)==4){
                    up=state_value[3][4];
                    down=state_value[3][4];
                    left=state_value[3][4];
                    right=state_value[3][4];
                    up_reward = 5;
```

```c
                    down_reward = 5;
                    left_reward = 5;
                    right_reward = 5;
            }

            next_state_value[i+1][j+1]= //write in the new state value array
                (0.25*1*(up_reward+0.9*up))+
                (0.25*1*(down_reward+0.9*down))+
                (0.25*1*(left_reward+0.9*left))+
                (0.25*1*(right_reward+0.9*right));
        }
    }
    for(int i=0;i<5;i++) //update the old state value array
        for(int j=0;j<5;j++)
            state_value[i+1][j+1]=next_state_value[i+1][j+1];
}
/**********************************************************************************
       Name:  write_file().
   Function:  Write array into CSV file.
 Parameters:  file_name.
**********************************************************************************/
void write_file(char * file_name){
    FILE *fpw;
    fpw = fopen(file_name, "wb");
    if (!fpw) printf("%s file create fail...\n",file_name);
    fprintf(fpw, "Iteration times:%d\n",count);
    for(int i=1;i<6;i++)
            for(int j=1;j<6;j++){
                fprintf(fpw, "%.1f,",state_value[i][j]);
                if(j==5) fputc('\n', fpw);
            }
    fclose(fpw);
}
```