实验题目：

传教士与野人

实验代码：

#include <stdio.h>

#include <string.h>

#define STEP\_MAX 20 //来回过河的次数

#define KIND\_NUM 3 //每个种类的数量

#define BOAT\_NUM 2 //船的载重量

typedef enum

{

BOAT\_THIS,//船在本岸

BOAT\_THAT,//船在对岸

} boat\_t;

typedef enum

{

ROAD\_GO,//过河

ROAD\_COME,//回来

} road\_t;

typedef struct

{

int ye;//对岸野人数量

int man;//对岸传教士数量

boat\_t boat;//船是否在对岸

} state\_t;//一种局面

typedef struct

{

int ye;//野人过河数量

int man;//传教士过河的数量

road\_t road;//回来或过河

} step\_t;//一个步骤

state\_t states[STEP\_MAX]={0};

step\_t steps[STEP\_MAX]={0};

//判断所有的野人和传教士是否都到了对岸

int final(state\_t s)

{

const state\_t cs=

{

KIND\_NUM,

KIND\_NUM,

BOAT\_THAT

};

if(memcmp(&cs,&s,sizeof(state\_t))==0)

{

return 1;

}

return 0;

}

//是否会发生野人杀传教士

int bad(state\_t s)

{

if(((KIND\_NUM-s.ye)>(KIND\_NUM-s.man) && (KIND\_NUM-s.man)>0)||(s.ye>s.man && s.man>0))

{

return 1;

}

return 0;

}

//第n种局面是否跟前面的相重复

int repeat(state\_t s[],int n)

{

int i;

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

{//已经有这种情况了

if (memcmp(&states[i], &states[n], sizeof(states[i])) == 0)

{

return 1;

}

}

return 0;

}

void output(step\_t steps[STEP\_MAX],int n)

{

char \*kinds[KIND\_NUM]={"野人","传教士"};

char \*routine[2]={"过河.","回来."};

int i;

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

{

if((steps[i].ye \* steps[i].man)>0)

{

printf("%d个%s和%d个%s%s\n",steps[i].ye,kinds[0],

steps[i].man,kinds[1],routine[steps[i].road]);

}

else if(steps[i].ye>0)

{

printf("%d个%s%s\n",steps[i].ye,kinds[0],

routine[steps[i].road]);

}

else if(steps[i].man>0)

{

printf("%d个%s%s\n",steps[i].man,kinds[1],

routine[steps[i].road]);

}

}

printf("\n");

}

//搜索过河方案(n表示过河次数)

void search(int n)

{

int i,j;

if(n>STEP\_MAX)

{//步数限制太少了

printf("Step Overflow!");

return;

}

if(final(states[n]))

{//都到对岸了

output(steps,n);

return;

}

if(bad(states[n]))

{//发生了野人杀传教士了

return;

}

if(repeat(states,n))

{//与前面的局面相重复了

return;

}

if(states[n].boat==BOAT\_THIS)

{//船在本岸

for (i = 0; i <= BOAT\_NUM && i<=(KIND\_NUM-states[n].ye); i++)

for (j = 0; j <= BOAT\_NUM-i && j<=(KIND\_NUM-states[n].man); j++)

{

if(i==0 &&j==0)

{

continue;

}

steps[n].ye=i;

steps[n].man=j;

steps[n].road=ROAD\_GO;

memcpy(&states[n+1], &states[n], sizeof(state\_t));

states[n+1].ye+=i;

states[n+1].man+=j;

states[n+1].boat=BOAT\_THAT;

search(n+1);

}

}

else

{

for (i = 0; i <= BOAT\_NUM && i<=states[n].ye; i++)

for (j = 0; j <= BOAT\_NUM-i && j<=states[n].man; j++)

{

if(i==0 &&j==0)

{

continue;

}

steps[n].ye=i;

steps[n].man=j;

steps[n].road=ROAD\_COME;

memcpy(&states[n+1], &states[n], sizeof(state\_t));

states[n+1].ye-=i;

states[n+1].man-=j;

states[n+1].boat=BOAT\_THIS;

search(n+1);

}

}

}

int main()

{

search(0);

return 0;

}

运行结果：

