***Artificial Intelligence Lab File***

***COE-408***

|  |
| --- |
| Submitted by:  *MANISH KUMAR GUPTA*  *COE-I*  *260/CO/08* |

***CONTENTS***

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Topic | Date | Signature |
|  |  |  |  |

**Water Jug Problem**

In this problem, two jugs are given and we have to get the desired amount of water in one of the jugs.

#include<iostream>

using namespace std;

int gcd(int,int);

int main()

{

int A,B,V,x,y,z;//A,x smaller

x=0,y=0,z=0;

cout<<"Enter size of Jug A(smaller in size): ";

cin>>A;

cout<<"Enter size of Jug B: ";

cin>>B;

if((z=gcd(A,B))!=1)

cout<<"Enter Volume required (multiple of "<<z<<"): ";

else

cout<<"Enter volume required: ";

cin>>V;

cout<<"\n A:0 B:0 \n";

if(gcd(z,V)==z)

while((x!=V)&&(y!=V))

{

if((x==0))

x=A;

else if((x==A)&&(y!=B))

{

z=(B-y);

if(z>A)

y+=x,x=0;

else

x-=z,y+=z;

}

else if((x!=A)&&(y!=B))

{

z=(B-y);

if(z>A)

y+=x,x=0;

else

x-=z,y+=z;

}

else if((y==B))

y=0;

cout<<"\n A:"<<x<<" B:"<<y;

}

cin.get(); cin.get();

return 0;

}

int gcd(int A,int B)

{

if(A==1)

return 1;

if(B%A!=0)

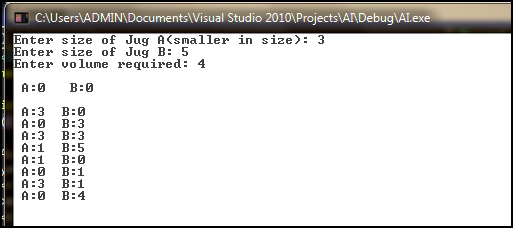
return gcd(B%A,A);

if(B%A==0)

return A;

return 0;

}



**Coins Problem**

In this problem, some types of coins are given and we have to get the desired amount of currency from them. Solution provides no. of solutions to the problem.

#include<iostream>

using namespace std;

int main()

{

int i,j,k,l,c=0,z=0,t;

for(i=0;i<500/25;i++)

for(j=0;j<500/50;j++)

for(k=0;k<500/100;k++)

for(l=0;l<500/200;l++,z++)

{

t=i\*25+j\*50+k\*100+l\*200;

if((t==500))

c++;

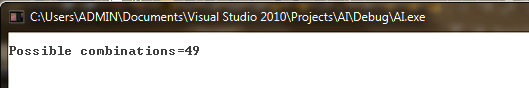
}

cout<<"\nc="<<c<<" z="<<z;

cin.get();

return 0;

}



**Problem**

There are a man with a wolf, goat and cabbage on the left bank of the river. There is a boat large enough to carry the man and only one of the other three. The man and his entourage wish to cross to the right bank, and the man can carry each across, one at a time. However, if the man leaves the wolf and goat unattended on either shore, the wolf will surely eat the goat. Similarly, if the goat and cabbage are left unattended, the goat will eat the cabbage. Is it possible to cross the river without eating the goat or cabbage being eaten?

//Man, Cabbage, Goat, Wolf

#include<iostream>

using namespace std;

void boat();

void print();

bool man, cab, goat, wolf;

int main()

{

man = cab = goat = wolf = false;

cout<<"\n Left Side ---- Right Side";

boat();

cin.get();

return 0;

}

void boat()

{

print();

if(man&&cab&&goat&&wolf)

return ;

if(!man)

{

if(!cab&&!goat&&!wolf)

goat=true;

else if(!wolf)

wolf=true;

else if(!cab)

cab=true;

else goat=true;

man=true;

boat();

}

else

{

if(goat&&(cab||wolf))

goat=false;

man=false;

boat();

}

}

void print()

{

char str1[5],str2[5];

str1[4]='\0';

str2[4]='\0';

man?str1[0]='M':str1[0]=' ';

cab?str1[1]='C':str1[1]=' ';

goat?str1[2]='G':str1[2]=' ';

wolf?str1[3]='W':str1[3]=' ';

!man?str2[0]='M':str2[0]=' ';

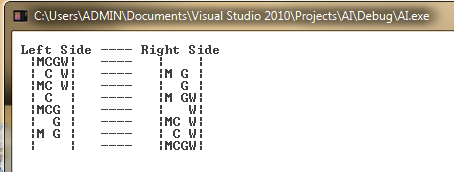
!cab?str2[1]='C':str2[1]=' ';

!goat?str2[2]='G':str2[2]=' ';

!wolf?str2[3]='W':str2[3]=' ';

cout<<"\n |"<<str2<<"| ---- |"<<str1<<"| ";

}



**N-Queen Problem**

In this problem, we have to place N no. of Queens on an NxN chess-board (N>3).

#include<iostream>

#include<cmath>

using namespace std;

#define SIZE 8

bool call(int);

bool chk(int, int);

int B[SIZE];

int main()

{

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

B[i]=-1;

cout<<"Queen Positions on Board --\n";

if(call(0)==true)

{

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

cout<<B[i]<<" ";

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

{

cout<<"\n-";

for(int z=0;z<SIZE;z++)

cout<<"--";

cout<<"\n";

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

if(B[i]==j)

cout<<"|Q";

else

cout<<"| ";

cout<<"|";

}

}

else

cout<<"ERROR";

cout<<"\n-";

for(int z=0;z<SIZE;z++)

cout<<"--";

cin.get();

return 0;

}

bool call(int q)

{

if(q==SIZE)

return true;

for(int c=0;c<SIZE;c++)

{

if((chk(q,c)==true)&&(call(q+1)==true))

return true;

}

return false;

}

bool chk(int q, int c)

{

int i,ci;

if(q==0)

{

B[q]=c;

return true;

}

else if(q==SIZE)

return false;

else

{

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

{

ci=B[i];

if(abs(q-i)==abs(c-ci))

return false;

if((c-ci)==0)

return false;

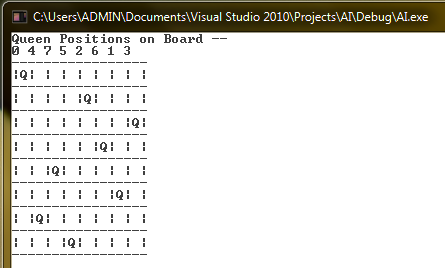
}

}

B[q]=c;

return true;

}



**Turing Machine**

Construct a Turing machine to accept Language L={0n1n} for n>1.

#include<iostream>

using namespace std;

int main()

{

enum state {S0,S1,S2,S3,S4};

state s;

int i=0,acc;

char str[100];

s=S0;

cout<<"Enter String";

cin>>str;

acc=1;

while(acc==1)

switch(s)

{

case S0:

if(str[i]=='0')

{

str[i]='x';

i++;

s=S1;

}

else if(str[i]=='y')

{

i++;

s=S3;

}

break;

case S1:

if(str[i]=='0'||str[i]=='y')

{

i++;

s=S1;

}

else if(str[i]=='1')

{

str[i]='y';

i--;

s=S2;

}

break;

case S2:

if(str[i]=='0'||str[i]=='y')

{

i--;

s=S2;

}

else if(str[i]=='x')

{

i++;

s=S0;

}

break;

case S3:

if(str[i]=='y')

{

i++;

s=S3;

}

else if(str[i]=='\0')

s=S4;

break;

case S4:

cout<<"String Accepted";

acc=0;

break;

}

cin.get();

cin.get();

return 0;

}

