ASDE Algorithm Test

Imagine Abhimanyu in Chakravyuha. There are 11 circles in the Chakravyuha surrounded by different enemies.

Abhimanyu is located in the innermost circle and has to cross all the 11 circles to reach Pandavas army back.

Given:

1. Each circle is guarded by different enemy where enemy is equipped with k1, k2……k11 powers.
2. Abhimanyu start from the innermost circle with p power Abhimanyu has a boon to skip fighting enemy a times.
3. Abhimanyu can recharge himself with power b times.
4. Battling in each circle will result in loss of the same power from Abhimanyu as the enemy.
5. If Abhimanyu enter the circle with energy less than the respective enemy, he will lose the battle.
6. k3 and k7 enemies are endured with power to regenerate themselves once with half of their initial power and can attack Abhimanyu from behind if he is battling in iteratively next circle.

Write an algorithm to find if Abhimanyu can cross the Chakravyuha and test it with two sets of test cases.

# Solution:

|  |
| --- |
| #include<iostream>  using namespace std;  int circles(float p,float a,float b,float k[]){  float power = p;  float temp;  for(int i=0;i<11;i++){  temp=k[i];  if(p<k[i] && a>0){  temp=0;  a--;  }  else if(p<k[i] && b>0){  p=power;  b--;  }  if(i==2 || i==6){  k[i+1]=k[i+1]+k[i]/2;  }    cout<<"Inner circle :"<<i+1<<endl;  cout<<"Enemy power k[i] :"<<k[i]<<endl;  cout<<"Abhimanyu power before p :"<<p<<endl;  cout<<"Skip power a :"<<a<<endl;  cout<<"Recharge power b :"<<b<<endl;  p=p-temp;  cout<<"Abhimanyu power after fight p :"<<p<<endl<<endl;  temp=0;    if(p<0){  cout<<"Abhimanyu lost at circle : "<<i+1;  return -1;  }  }    return p;  }  int main(){  float abhimanyu\_power;  float skip;  float power\_recharge;  float enemy\_power[11]={10,20,30,40,50,60,70,80,90,100,110};    cout<<"Enter Abhimanyu power p :";  cin>>abhimanyu\_power;  cout<<"Enter Abhimanyu skip power a :";  cin>>skip;  cout<<"Enter Abhimanyu recharge power b :";  cin>>power\_recharge;  cout<<endl;    int isWon = circles(abhimanyu\_power,skip,power\_recharge,enemy\_power);    if(isWon>0){  cout<<"Abhimanyu won the fight ";  }    return 0;  } |

Output:

Enemy\_Power = [ k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11]

Enemy\_Power Consider as = [ 10,20,30,40,50,60,70,80,90,100,110 ]

# Test Case 1

|  |
| --- |
|  |

# Test Case 2

|  |
| --- |
|  |