```
#include<iostream>
using namespace std;
#include<set>
#include<vector>
#include<string>

int ar[5][2];
vector<int> combination;
int Fibonacci(int n)
{
    if(n==0)
        return 0;
    else if(n==1)
        return 1;
    else if(n>1)
        return Fibonacci(n-1)+Fibonacci(n-2);
    else
        return -1;
}
int path(int right, int down)
{
    int sum=0;
    //int index = 5 - (right + down);

    if(right == 0 && down==0 )
    {   //cout<<"-----"<<endl;
        //for(int i=0; i<5; i++)
            cout << ar[i][0]<<" "<< ar[i][1]<<endl;
        return 1;
    }
    if(right>0)
    {
        //ar[index][0] = ar[index-1][0];
        //ar[index][1] = ar[index-1][1] - 1;
        sum=sum + path(right-1, down);
    }
    if(down>0)
    {
        //ar[index][0] = ar[index-1][0]-1;
        //ar[index][1] = ar[index-1][1];
        sum=sum + path(right, down-1);
    }
    return sum;
}
```

```
void pretty_print(const vector<int>& v)
{
    static int count = 0;
    cout << "combination no " << (++count) << ": [ ";
    for (int i = 0; i < v.size(); ++i)
    {
        cout << v[i] << " ";
    }
    cout << "] " << endl;
}

void go(vector<int> people, int offset, int k)
{
    if (k == 0)
    {
        pretty_print(combination);
        return;
    }
    for(int i=offset; i<=people.size()-k; i++)
    {
        combination.push_back(people[i]);
        go(people, i+1,k-1);
        combination.pop_back();
    }
}
void allsubset(vector<int> myset, int com)
{
    /*
    if(com>myset.size())
        return;

    go(myset,0, com); //com-lu combinatlari print et
    allsubset(myset, com+1);
    */
    for(; com<=myset.size(); com++)
        go(myset,0, com);
}
```

```
void swap(char *x, char *y)
{
   char temp;
   temp = *x;
   *x = *y;
   *y = temp;
}
void permutation(char* front, int left, int right)
{
     if(left==right)
          printf("%s\n", front);
     else
     {
          for(int i=left; i<=right; i++)
          {
               swap((front+left),(front+i));
               permutation(front, left+1,right);
               swap((front+left),(front+i));
          }
     }
}

void printParanthesis(int l, int r, char str[],int count)
{
     if(l==0 && r==0)
     {
          cout<<str<<endl;
          return;
     }
     if(l>0)
     {
          str[count]='(';
          printParanthesis(l-1,r,str,count+1);
     }
     if(r>l)
     {
          str[count]=')';
          printParanthesis(l,r-1,str,count+1);
     }

}
```

```
int cents2(int n, int denom)
{
     int next_den = 0;

          if(denom==25)
               next_den=10;
          else if(denom==10)
               next_den=5;
          else if(denom==5)
               next_den=1;
          else if(denom==1)
               return 1;

          int ways = 0;
          for(int i=0;i*denom<=n;i++)
          {
               ways += cents2(n - i * denom, next_den);
          }
          return ways;
}
int main()
{
//    cout<<Fibonacci(5)<<endl;
//    arraya[0][0]=2;
//    arraya[0][1]=2;
//    cout<<path(2,2)<<endl;

//set<int>myset;
//    for(int i=0; i<8; i++)
//         myset.insert(i);

vector<int>myset;
     for(int i=0; i<3; i++)
          myset.push_back(i);

//    allsubset(myset, 1);

//    string myname="bat";
//    char* front=&myname[0];
//    permutation(front,0,2);

//    char str[3*2];
//    printParanthesis(3,3,str,0);

     cout<<cents(1,25)<<endl;
     cout<<cents(5,25)<<endl;
     cout<<cents(10,25)<<endl;
     //cout<<cents(25,25)<<endl;}
```