/*Fibonacci search is one more popular searching technique. This searching technique is similar to the binary searching technique.

Similarities between fibonacci search and binary search :

1. The precondition is that the elements of the array should be sorted.
2. Both are divide and conquer techniques.
3. The time complexity of both the algorithms is O(log n)*/

```
//Created By Ritwik Chandra Pandey on 8th Nov 2021
//Fibonacci Search


#include <stdio.h>
#include <conio.h>

/* Function to find the minimum of x and y */
int min(int x, int y) {
	if(x>=y){
		return y;

	}else{
		return x;
	}
}
/* If element x is not present in the arr[] of size n return -1 , else return the index */
int fibonaccianSearch(int arr[], int x, int n) {
	int m,m1,m2,offset;
	m2 = 0;
	m1 = 1;
	m = m1+m2;
	while(m<n){
		m2 = m1;
		m1 = m;
		m = m1+m2;
	}
	offset = -1;
	while(m>1){
		int i=min(offset+m2,n-1);
		if(arr[i]<x){
			m = m1;
			m1 = m2;
```

```
                        m2 = m-m1;
                        offset = i;
                }else if(arr[i]>x){
                        m = m2;
                        m1 = m1-m2;
                        m2 = m-m1;
                }
                  else
                     return i;
        }
        if(m1!=0 && arr[offset+1]==x){
                return offset+1;
        }
        return -1;
}

int main() {
        int size;
        int *arr, i,x,result=-1;
        printf("Enter the size of an array: ");
    scanf("%d",&size);
        arr = (int*) malloc(size * sizeof(int));
        printf("Enter the %d array elements\n",size);
    for (i = 0; i < size; i++) {
                scanf("%d", &arr[i]);
    }
    printf("Enter the element to be searched: ");
    scanf("%d",&x);
    result = fibonaccianSearch(arr,x,size+1);
        if (result != -1 )
                printf("Element found at index: %d.\n",result);
        else
                printf("Element not found.\n");
        return 0;
}
```