

```

/*
Εξηγήστε τι κάνει η ακόλουθη συνάρτηση και δώστε μία υλοποίηση
χωρίς αναδρομή:
int test(const int b[], int start, int end, int x)
{
    if (start > end) return -1;
    else if (x==b[start]) return start;
    else return test(b, start+1, end, x);
}

```

Υλοποιήστε κυρίως πρόγραμμα που γεμίζει τον πίνακα με θετικά στοιχεία (αμυντικός προγραμματισμός) και καλεί την υλοποιημένη από εσάς συνάρτηση.

```

#include <stdio.h>
#define N 10

int test1(const int b[], int start, int end, int x)
{
    if (start > end) return -1;
    else if (x==b[start]) return start;
    else return test1(b, start+1, end, x);
}

int test2(const int b[], int start, int end, int x)
{
    int i;
    for (i=start; i<=end; i++)
        if (x==b[i]) return i;
    return -1;
}

int main()
{
    int data[N];
    int element;
    int thesi;

    int i;
    for (i=0; i<N; i++)
    {
        do
        {
            printf("Give %d number (>=0: ", i);
            scanf("%d", &data[i]);
        }
        while (data[i]<0);
    }
}

```

```
printf("Give element to search for: ");
scanf("%d", &element);

thesi=test1(data,0, N-1, element);
if (thesi==-1) printf("The element does not exist.");
else printf("The element exist in %d position.\n", thesi);

thesi=test2(data, 0, N-1, element);
if (thesi==-1) printf("The element does not exist.");
else printf("The element exist in %d position.\n", thesi);

return 0;
}
```