Homework 3 - esercizio 1

- Caso base: il valore corrente in input è -1; salvo -1 nella cella corrente del vettore e termino la ricorsione.

Homework 3 - esercizio 1

- Caso base: il valore corrente in input è -1; salvo -1 nella cella corrente del vettore e termino la ricorsione.
- Caso limite: la prima cella del vettore deve semplicemente contenere il valore in input.

Homework 3 - esercizio 1

- Caso base: il valore corrente in input è -1; salvo -1 nella cella corrente del vettore e termino la ricorsione.
- Caso limite: la prima cella del vettore deve semplicemente contenere il valore in input.
- Caso generale: nella cella i-1 ho il valore della somma parziale precedente. Salvo nella cella i il valore della cella i-1 sommato al valore corrente.

Homework 3 - esercizio 1

```
#include <stdio.h>
#define N 100
void somma_prefissi (long[],
                             int);
int main (void ) {
  long v [N];
  int i=0;
  somma_prefissi(v, 0);
  while (1) {
    if (v[i] != -1) {
       printf("%ld ", v[i]);
       i++;
    } else {
      break;
    }
  }
  printf("\n");
}
```

Homework 3 - esercizio 1

```c
#include <stdio.h>
#define N 100
void somma_prefissi (long[],
                     int);
int main (void ) {
  long v [N];
  int i=0;
  somma_prefissi(v, 0);
  while (1) {
    if (v[i] != -1) {
       printf("%ld ", v[i]);
       i++;
    } else {
      break;
    }
  }
  printf("\n");
}
```

```c
void somma_prefissi(long v[],
                    int i){
  scanf("%ld", &v[i]);
  if (v[i]==-1){
    return;
  }
  if (i>0){
    v[i]+= v[i-1];
  }
  somma_prefissi(v, i+1);
}
```

Homework 3 - esercizio 2

- Caso base: il valore corrente in input è -1; salvo -1 nella cella corrente del vettore e termino la ricorsione, restituendo il valore 0.

Homework 3 - esercizio 2

- Caso base: il valore corrente in input è -1; salvo -1 nella cella corrente del vettore e termino la ricorsione, restituendo il valore 0.
- Caso generale: ottengo il valore della somma dei suffissi dalla chiamata ricorsiva, lo sommo al valore corrente, assegno il valore della somma alla cella i e lo restituisco alla funzione chiamante.

Homework 3 - esercizio 2

```
#include <stdio.h>
#define N 100
long somma_suffissi (long[],
                     int);
int main (void) {
  long v [N];
  int i=0;
  somma_suffissi(v, 0);
  while (1) {
    if (v[i] != -1) {
      printf("%ld ", v[i]);
      i++;
    } else {
      break;
    }
  }
  printf("\n");
}
```

Homework 3 - esercizio 2

```c
#include <stdio.h>
#define N 100
long somma_suffissi (long[],
                     int);
int main (void) {
  long v [N];
  int i=0;
  somma_suffissi(v, 0);
  while (1) {
    if (v[i] != -1) {
      printf("%ld ", v[i]);
      i++;
    } else {
      break;
    }
  }
  printf("\n");
}
```

```c
long somma_suffissi(long v[],
                    int i){
  scanf("%ld", &v[i]);
  if (v[i]==-1){
    return 0;
  }
  v[i]+= somma_suffissi(v, i+1);
  return v[i];
}
```

Homework 3 - esercizio 3

- Caso base della funzione ricorsiva: il valore corrente è -1; termino la ricorsione.
- Caso generale: ipotizziamo di aver ricevuto i conteggi correnti delle occorrenze al momento della chiamata; se il valore corrente è uguale a k,
  1. se mi trovo nella prima cella o il valore precedente è minore di k incremento le occorrenze di k precedute da un valore minore.

Homework 3 - esercizio 3

- Caso base della funzione ricorsiva: il valore corrente è -1; termino la ricorsione.
- Caso generale: ipotizziamo di aver ricevuto i conteggi correnti delle occorrenze al momento della chiamata; se il valore corrente è uguale a k,
  1. se mi trovo nella prima cella o il valore precedente è minore di k incremento le occorrenze di k precedute da un valore minore.
  2. se il valore successivo è maggiore di k o uguale a -1 incremento le occorrenze di k seguite da un valore maggiore.

Homework 3 - esercizio 3

- Caso base della funzione ricorsiva: il valore corrente è -1; termino la ricorsione.
- Caso generale: ipotizziamo di aver ricevuto i conteggi correnti delle occorrenze al momento della chiamata; se il valore corrente è uguale a k,
  1. se mi trovo nella prima cella o il valore precedente è minore di k incremento le occorrenze di k precedute da un valore minore.
  2. se il valore successivo è maggiore di k o uguale a -1 incremento le occorrenze di k seguite da un valore maggiore.

  indipendentemente dal valore corrente (che è comunque diverso da -1) eseguo la chiamata ricorsiva, passando alla cella successiva del vettore e fornendo i valori aggiornati dei contatori delle occorrenze.

Homework 3 - esercizio 3

```
#include <stdio.h>
#define N 100
void occorrenze ( int v [],
  int k, int i, int * occ_min,
                int * occ_max);
int main (void ) {
  int v [N];
  int i=0, k, occ_min, occ_max;
  scanf("%d", &k);
  if (k<0) return;
  while (1) {
    scanf("%d", &v[i]);
    if (v[i] == -1) break;
    i++;
  }
  occorrenze(v, k, 0, &occ_min,
                      &occ_max);
  printf("%d %d\n", occ_min,
                       occ_max);
}
```

Homework 3 - esercizio 3

```c
#include <stdio.h>
#define N 100
void occorrenze ( int v [],
  int k, int i, int * occ_min,
                int * occ_max);
int main (void ) {
  int v [N];
  int i=0, k, occ_min, occ_max;
  scanf("%d", &k);
  if (k<0) return;
  while (1) {
    scanf("%d", &v[i]);
    if (v[i] == -1) break;
    i++;
  }
  occorrenze(v, k, 0, &occ_min,
                      &occ_max);
  printf("%d %d\n", occ_min,
                      occ_max);
}
```

```c
void occorrenze ( int v [],
     int k, int i, int * occ_min,
                   int * occ_max ){
  if (i==0){
    *occ_min = 0;
    *occ_max = 0;
  }
  if (v[i]==-1 || k<0) return;
  if (v[i]==k){
    if(i==0 || v[i-1] < k)
                   (*occ_min)++;
    if(v[i+1]>k || v[i+1]==-1)
                   (*occ_max)++;
  }
  occorrenze(v, k, i+1, occ_min,
                        occ_max);
}
```

Tipi di dati struct

Il comando typedef e la parola chiave struct permettono di definire dei tipi di dati personalizzati, composti da più campi di tipo diverso.

```
typedef struct moto {
  char marca [20];
  char nome [20];
  int cc;
  float km_litro;
} Moto;
...
Moto sv;
sv.marca[0] = 'S'; //uzuki\0
....
sv.nome[0] = 'S';//V 1000\0
....
sv.cc = 996;
sv.km_litro = 10.5 //andando piano :(
```

Esercizio

Scrivere una funzione di ordinamento per un vettore i cui elementi sono coppie di valori interi: un valore `min` e uno `max`.
L'ordinamento deve essere non decrescente in base al valore della media fra `min` e `max`.

Suggerimento: scrivere una funzione per effettuare il confronto e una funzione per effettuare lo swap fra due celle, modificate la vecchia funzione randomize per popolare il vostro vettore.

Esercizio

Dato un vettore i cui elementi sono un carattere e un vettore di interi che rappresentano le posizioni in cui il carattere deve comparire, scrivere una funzione che stampi la stringa corrispondente.
Esempio:
'a' {1,3,5,-1}
'b' {0,-1}
'n' {2,4,-1}
'\0' {-1}
come vettore, deve stampare "banana"