BIAGIO SALZILLO M63001227

1)

#include <stdio.h>

#include <stdlib.h>

int ricerca(int \*vett, int start, int end, int val);

void stampa\_vett(int \*vett, int n);

int main(void) {

// your code goes here

int \*vett;

int n, t, val;

scanf("%d", &t);

for(int k = 0; k < t; k++){

scanf("%d %d", &val, &n);

vett = (int \*)malloc(sizeof(int)\*n);

for(int i =0; i < n; i++){

scanf("%d", &vett[i]);

}

printf("%d\n", ricerca(vett, 0, n-1, val));

}

return 0;

}

void stampa\_vett(int \*vett, int n){

for(int i = 0; i < n; i++){

printf("%d ", vett[i]);

}

}

int ricerca(int \*vett, int start, int end, int val){

int c1, c2;

if(start == end){

if(vett[start] == val) return 1;

return 0;

} //caso in cui ho un elemento

if(end > start){

int mid = (start + end)/2;

c1 = ricerca(vett, start, mid, val);

c2 = ricerca(vett, mid + 1, end, val);

return c1 + c2;

}

}

**COMPLESSITA: O(n)**

2)

#include <stdio.h>

#include <stdlib.h>

void ricerca\_seq(int s, int n, int p, int i, int j, int k, int \*vett);

void v\_primi(int \*vett, int n);

int main(void) {

// your code goes here

int t, s, n, p;

int \* vett;

scanf("%d", &t);

for(int i = 0; i < t; i++){

scanf("%d %d %d", &s, &n, &p);

vett = (int \*)malloc(sizeof(int)\*s);

v\_primi(vett, s); //genero il vettore di numeri primi fino a s

printf("CASO TEST %d", t+1);

ricerca\_seq(s, n, p, 0, 1, 2, vett);

}

return 0;

}

void ricerca\_seq(int s, int n, int p, int i, int j, int k, int \* vett){

if(n == 1) printf("\n %d", s);

if(s == vett[i] + vett[j] + vett[k] && vett[i] != vett[j] && vett[k] != vett[i] && vett[j]!=vett[k] && vett[i]>p && vett[j]>p && vett[k]>p){

printf("%d %d %d\n", vett[i], vett[j], vett[k]);

return;

}

if(s > vett[i] + vett[j] + vett[k]){

ricerca\_seq(s, n, p, i, j, k + 1, vett);

ricerca\_seq(s, n, p, i, j+1, k, vett);

ricerca\_seq(s, n, p, i+1, j, k, vett);

}

}

void v\_primi(int \*v, int N){

int tmp[N];

int k =0;

for(int i =0; i<N; i++) tmp[i] = 0;

int j;

for(unsigned int i = 2; i <= N; ++i)

{

if(!tmp[i])

{

//printf("%d\n", i);

v[k] = i;

k++;

for(j = 2 \* i; j <= N; j += i)

{

tmp[j] = 1;

}

}

}

}

COMPLESSITA: FATTORIALE