

**Algoritam**

Neka imamo skup elemenata  $\{a_1, a_2, \dots, a_n\}$  i želimo napraviti varijacije sa ponavljanjem  $k$ -te klase. Prva varijacija će biti  $a_1 a_1 \dots a_1 a_1$  ( $a_1 - k$  puta). Generisanje sledeće varijacije se radi tako što se element sa poslednje pozicije upoređuje sa poslednjim elementom u nizu, tj. gleda se da li je taj element stigao do zadnje vrednosti (u ovom slučaju  $a_n$ ). Ukoliko nije, element na zadnjoj poziciji se povećava za 1, tj. uzima vrednost sledećeg elementa u skupu pa je sledeća varijacija  $a_1 a_1 \dots a_1 a_2$ . Ukoliko je poslednji element došao do poslednjeg u skupu tj. imamo varijaciju  $a_1 a_1 \dots a_1 a_n$  tada tražimo u levom podnizu vrednost koja nije stigla do maksimuma niza. U slučaju da postoji taj element se postavlja se povećava za 1, tj. uzima vrednost sledećeg elementa u skupu, a desni deo niza se postavlja na najmanji element u skupu ( $a_1$ ).

U ovom slučaju se povećava element na preposlednjoj poziciji tj. imamo da je sledeća varijacija  $a_1 a_1 \dots a_2 a_1$ . U slučaju da ne postoji element u levom podnizu da je manji od maksimuma skupa, tada smo stigli do varijacije  $a_n a_n \dots a_n a_n$  i time završavamo pravljenje varijacija.

**Program**

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int NadjiIndex(int* niz,int max,int k)
```

```
{
    while(k--)
        if(max>niz[k])
            return k;
    return k;
}
```

```
/*Funkcija koja vraća indeks niza čija je vrednost manja od maksimuma,
pretraga se vrši od kraja ka početku, a u koliko ne postoji takav element
vraća se -1*/
```

```
void PostaviNaNulu(int* niz,int max,int start)
```

```
{
    while(max>start)
        niz[--max]=0;
}
```

```
/*Funkcija koja postavlja na nulu sve elemente od kraja niza do određenog
indeksa*/
```

```
void MakeFirst(int* niz,int k)
```

```
{
    while(k--)
        niz[k]=0;
}
```

```
/*Funkcija koja postavlja sve elemente niza počev od zadnjeg na 0*/
```

```
void Odstampaj(int* skup,int* niz,int k)
```

```
{
    int i;

    for(i=0;i<k;i++)
```

```
    printf("%d",skup[niz[i]]);
    putchar('\n');
}
/*Funkcija koja stampa elemente skupa sa indeksima vrednosti elemenata niza*/
```

```
void NapraviSledecu(int* niz,int k,int start)
{
    niz[start]++;
    PostaviNaNulu(niz,k,start+1);
}
/*Funkcija koja uvecava odredjeni element niza za 1 i poziva funkciju
PostaviNaNulu*/
```

```
void UnosElemenata(int* skup,int n)
{
    int i;

    for(i=0;i<n;i++)
    {
        printf("\nUnesite %d. element : ",i+1);
        scanf("%d",&skup[i]);
    }
}
/*Funkcija za unosenje elemenata skupa*/
```

```
int main()
{
    int *niz,k,start,n,broj=0,*skup;

    printf("Unesite broj elemenata : ");
    scanf("%d",&n);
    //Unosi se broj elemenata skupa

    skup=(int *)malloc((n-1)*sizeof(int));//Alociramo memoriju za skup

    if(niz==NULL|skup==NULL)
    {
        printf("Greska!");
        exit(0);
    }
    UnosElemenata(skup,n);//Unosimo elemente
    printf("\nUnesite k : "); scanf("%d",&k); //Unosenje klase varijacija
    niz=(int *)malloc((k-1)*sizeof(int));//Alociramo memoriju za niz
    MakeFirst(niz,k);//Pravljenje prve varijacije
    while(++broj)
    {
        Odstampaj(skup,niz,k);//Stampanje varijacije
    }
}
```

```
if((start=NadjiIndex(niz,n-1,k))>=0)
/*Nalazenje indeksa niza koji odgovara uslovu da je manji od maksimuma i provera da li je veci od -1*/
  NapraviSledecu(niz,k,start);//Pravljenje sledece varijacije
else
  break;//Ukoliko smo stigli do poslednje varijacije izlazimo iz petlje
}
printf("Ukupno ih ima : %d\n",broj);//Stampanje broja varijacija
free(&skup);//Oslobadjanje memorije
free(&niz);//Oslobadjanje memorije
return 0;
}
```

**Primer**

Skup : { 2, 1, 4 }, k=3

Rezultat :

222	122	422
221	121	421
224	124	424
212	112	412
211	111	411
214	114	414
242	142	441
241	141	442
244	144	444

Ukupno ih ima : 27