



Найти k -ый по величине (сверху) элемент массива

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <ctime>
#define SIZE 10
using namespace std;
int ms[SIZE];
void printmass(int sm[], int razmer)
{
    int i;
    for (i=0; i<razmer ;i++)
    {
        printf("%d ",sm[i]);
    }
    printf("\n");
}
void fillmassrandom(int sm[], int razmer)
{
    int i,znak;
    for (i=0; i<razmer ;i++)
    {
        znak=rand()%2;
        if (znak==0)
        {
            sm[i]=rand()%20;
        }
        else
        {
            sm[i]=(-1)*rand()%10;
        }
    }
}
void fillmass2(int sm[], int razmer, int k)
{
    clock_t start=clock();
    int i,h=0,j,temp;
    int* help=(int *)calloc(k+1,sizeof(int));
    i=0;
    while (i<k+1)
    {
        help[i]=0;
        i++;
    }
    i=0;
    while (i<razmer)
    {
        if(h<k+1)//добавление без вытеснений
        {
            help[h]=sm[i];
            //начинаем пропихивать
            j=h;
            while(j>0 && help[j]<help[j-1])
            {
                temp=help[j];
                help[j]=help[j-1];
                sm[j-1]=temp;
                j--;
            }
            h++;
        }
        else//добавление с вытеснениями
        {
            //сдвиг вправо с занулением последнего
            j=1;
            while(j<k+1)
            {
                help[j-1]=help[j];
                j++;
            }
            help[k]=0;
            help[k]=sm[i];
            //начинаем пропихивать
            j=k;
            while(j>0 && help[j]<help[j-1])
            {
                temp=help[j];
                help[j]=help[j-1];
                help[j-1]=temp;
                j--;
            }
        }
        i++;
    }
    printf("%d\n",help[1]);
    clock_t finish=clock();
    printf("time=%d\n",finish-start);
}
int main()
{
    srand(time(NULL));
    int ms[SIZE]={8,-8,-8,12,-12,12,12,-13,14,-15};
    fillmassrandom(ms,SIZE);
    printmass(ms,SIZE);
    fillmass2(ms,SIZE,5);
    printmass(ms,SIZE);
}
```

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <ctime>
#define SIZE 10
using namespace std;
int ms[SIZE];
void printmass(int sm[], int razmer)
{
    int i;
    for (i=0; i<razmer ;i++)
    {
        printf("%d ",sm[i]);
    }
    printf("\n");
}
void fillmassrandom(int sm[], int razmer)
{
    int i,znak;
    for (i=0; i<razmer ;i++)
    {
        znak=rand()%2;
        if (znak==0)
        {
            sm[i]=rand()%20;
        }
        else
        {
            sm[i]=(-1)*rand()%10;
        }
    }
}
void fillmass2(int sm[], int razmer, int k)
{
    clock_t start=clock();
    int i,h=0,j,temp;
    int* help=(int *)calloc(k+1,sizeof(int));
    i=0;
    while (i<k+1)
    {
        help[i]=0;
        i++;
    }
    i=0;
    while (i<razmer)
    {
        if(h<k+1)//добавление без вытеснений
        {
            help[h]=sm[i];
            //начинаем пропихивать
            j=h;
            while(j>0 && help[j]<help[j-1])
            {
                temp=help[j];
                help[j]=help[j-1];
                help[j-1]=temp;
                j--;
            }
            h++;
        }
        else//добавление с вытеснениями
        {
            //сдвиг вправо с занулением последнего
            j=1;
            while(j<k+1)
            {
                help[j-1]=help[j];
                j++;
            }
            help[k]=0;
            help[k]=sm[i];
            //начинаем пропихивать
            j=k;
            while(j>0 && help[j]<help[j-1])
            {
                temp=help[j];
                help[j]=help[j-1];
                help[j-1]=temp;
                j--;
            }
        }
        i++;
    }
    printmass(help,k+1);
    printf("%d\n",help[1]);
    clock_t finish=clock();
    printf("time=%d\n",finish-start);
}
int main()
{
    srand(time(NULL));
    int ms[SIZE]={8,-8,-8,12,-12,12,12,-13,14,-15};
    fillmassrandom(ms,SIZE);
    printmass(ms,SIZE);
    fillmass2(ms,SIZE,5);
    printmass(ms,SIZE);
}
```

5 1 23 2 525 6 31 43 45 7

0 0 0 0 0

5 0 0 0 0

5 1 0 0 0 -> 1 5 0 0 0

1 5 23 0 0

1 5 23 2 0 -> 1 5 2 23 0->1 2 5 23 0

1 2 5 23 525

2 5 23 525 0 -> 2 5 23 525 6 -> 2 5 23 6 525->2 5 6 23 525

5 6 23 525 0 -> 5 6 23 525 31 -> 5 6 23 31 525

6 23 31 525 0->6 23 31 525 43->6 23 31 43 525

23 31 43 525 0->23 31 43 525 45->23 31 43 45 525

31 43 45 525 0->31 43 45 525 7->31 43 45 7 525->31 43 7 45 525>31 7 4345 525->7 31 43 45 525