acm.mipt.ru

олимпиады по программированию на Физтехе

Раздел «Язык Си» . SortProblem :

Задача сортировки

Сортировка пузырьком

Задача сортировки – одна из первых интересных и сложных задач теории алгоритмов. Один из простейших алгоритмов, решающих эту задачу, – это метод пузырька.

```
#include<stdio.h>
#define N 1000
int
main()
{
   int n, i;
   int a[N];
   scanf("%d", &n);
   for(i = 0 ; i < n; i++)
   {
       scanf("%d", &a[i]);
   }
   for(i = 0 ; i < n-1 ; i++)
   {
      for(j = i + 1 ; j < n ; j++)
      {
          if(a[i] > a[j])
          {
             int tmp = a[i]; a[i] = a[j] ; a[j] = tmp;
          }
       }
   }
   for(i = 0 ; i < n; i++)
   {
       printf("%d ", a[i]);
   }
   return 0;
}
```

Функция qsort из библиотеки stdlib

Два оператора for, в которых происходит сортировка можно заменить на одну строчку

```
qsort(a, n, sizeof(int), cmp );
```

– это функция, описанная в библиотеке stdlib.

Поэтому в начале программы нужно добавить

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

Четвертый аргумент функции qsort – это имя функции, которая умеет сравнивать два элемента массива. В нашем случае это

Поиск

Поиск

Раздел «Язык Си»

Главная
Зачем учить С?
Определения
Инструменты:
Поиск
Изменения
Index
Статистика

Разделы

Информация
Алгоритмы
Язык Си
Язык Ruby
Язык Ассемблера
El Judge
Парадигмы
Образование
Сети
Objective C

Logon>>

```
int cmp(const void *a, const void *b)
{
    return *(int*)a - *(int*)b;
}
```

Таким образом мы получили следующую программу

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

#define N 1000
int cmp(const void *a, const void *b)
{
    return *(int*)a - *(int*)b;
}

int
main()
{
   int n, i;
   int a[N];
   scanf("%d", &n);
   for(i = 0 ; i < n; i++)
   {
       scanf("%d", &a[i]);
   }

   qsort(a, n, sizeof(int), cmp );

   for(i = 0 ; i < n; i++)
   {
       printf("%d ", a[i]);
   }
   return 0;
}
```

Динамическое выделение памяти

Ниже приведена программа, где память под массив выделяется динамически:

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

#define N 1000
int cmp(const void *a, const void *b)
{
    return *(int*)a - *(int*)b;
}

int
main()
{
   int n, i;
   int *a;
   scanf("%d", &n);
   a = (int*) malloc(sizeof(int)*n);
   for(i = 0 ; i < n; i++)
   {
       scanf("%d", &a[i]);
   }

   qsort(a, n, sizeof(int), cmp );

   for(i = 0 ; i < n; i++)
   {
       printf("%d ", a[i]);
```

```
    }
    free(a);
    return 0;
}
```

malloc – это memory allocate, то есть "выдели память". Единственный аогумент этой функции – число байт, которое вам нужно. Всю память, которая была выделена с помощью malloc нужно в конце освобождать с помощью функции free. Аргумент функции free – это указатель на начало выделенной когда-то памяти.

Программа сортировки строк в алфавитном порядке

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#define N 100
#define M 30

int main(int argc, char* argv[])
{
   char a[N][M];
   int n,i;

   scanf("%d",&n);
   for (i=0;i<n;i++)
         scanf("%s",&a[i]);

   qsort(a, n, sizeof(char[M]), (int (*)(const void *,const void *)) strcmpi );

   for (i=0;i<n;i++)
      printf("%s\n",a[i]);

   return 0;
}
```

-- ArtemVoroztsov - 10 Nov 2004