Практическое занятие 8

Массивы

Пименов Евгений Сергеевич

Курс «Программирование»

Сибирский государственный университет телекоммуникаций и информатики (Новосибирск)

Осенний семестр, 2016

Повторение лекционного материала

- Что такое массив?

Повторение лекционного материала

- Что такое массив?
- Как элементы массива расположены в памяти?

Повторение лекционного материала

- Что такое массив?
- Как элементы массива расположены в памяти?
- Как осуществляется доступ к элементам массива?

Повторение лекционного материала

- Что такое массив?
- Как элементы массива расположены в памяти?
- Как осуществляется доступ к элементам массива?
- Как зависит время доступа к элементу от его индекса?

Массивы в языке C: инициализация


```
int array[] = {1, 2, 3, 4};
```

Массивы в языке C: инициализация


```
int array[] = {1, 2, 3, 4};

int array[5] = {1, 2, 3, 4};
```

Массивы в языке C: инициализация


```
int array[] = {1, 2, 3, 4};

int array[5] = {1, 2, 3, 4}; // 1, 2, 3, 4, 0
```

Массивы в языке C: инициализация


```
int array[] = {1, 2, 3, 4};

int array[5] = {1, 2, 3, 4}; // 1, 2, 3, 4, 0
```

Размер массива — константа этапа компиляции (C89).

Размер массива может быть переменной (C99).

Объявление массива


```
01. void make_leaf()
02. {
03.     // intentionally empty body
04. }
05.
06. int main()
07. {
08.     int array[10000];
09.     // don't ask why I'm here
10.     make_leaf();
11.     return 0;
12. }
```

Объявление массива


```
01. void make_leaf()
02. {
03.     // intentionally empty body
04. }
05.
06. int main()
07. {
08.     int array[10000];
09.     // don't ask why I'm here
10.     make_leaf();
11.     return 0;
12. }
```

```
01. main:
02. .LFB1:
03.         .cfi_startproc
04.         pushq   %rbp
05.         .cfi_def_cfa_offset 16
06.         .cfi_offset 6, -16
07.         movq    %rsp, %rbp
08.         .cfi_def_cfa_register 6
09.         subq    $40000, %rsp
10.         movl    $0, %eax
11.         call    make_leaf
12.         movl    $0, %eax
13.         leave
14.         .cfi_def_cfa 7, 8
15.         ret
16.         .cfi_endproc
```

Объявление массива


```
01. void make_leaf()
02. {
03.     // intentionally empty body
04. }
05.
06. int main()
07. {
08.     int array[10000];
09.     // don't ask why I'm here
10.     make_leaf();
11.     return 0;
12. }
```

```
01. main:
02. .LFB1:
03.         .cfi_startproc
04.         pushq   %rbp
05.         .cfi_def_cfa_offset 16
06.         .cfi_offset 6, -16
07.         movq    %rsp, %rbp
08.         .cfi_def_cfa_register 6
09.         subq    $40000, %rsp
10.         movl    $0, %eax
11.         call    make_leaf
12.         movl    $0, %eax
13.         leave
14.         .cfi_def_cfa 7, 8
15.         ret
16.         .cfi_endproc
```

Массивы переменной длины


```c
#include <stdlib.h>

void make_leaf()
{
    // intentionally empty body
}

int main()
{
    size_t n = 10000;
    int array[n];
    // don't ask why I'm here
    make_leaf();
    return 0;
}
```

```asm
main:
.LFB3:
  .cfi_startproc
  pushq %rbp
  .cfi_def_cfa_offset 16
  .cfi_offset 6, -16
  movq  %rsp, %rbp
  .cfi_def_cfa_register 6
  pushq %rbx
  subq  $40, %rsp
  .cfi_offset 3, -24
  movq  %rsp, %rax
  movq  %rax, %rbx
  movq  $10000, -40(%rbp)
  movq  -40(%rbp), %rax
  movq  %rax, %r8
  subq  $1, %r8
  movq  %r8, -32(%rbp)
  movq  %rax, %rsi
  movl  $0, %edi
  movq  %rax, %rdx
  movl  $0, %ecx
  salq  $2, %rax
  leaq  3(%rax), %rdx
  movl  $16, %eax
  subq  $1, %rax
  addq  %rdx, %rax
  movl  $16, %ecx
  movl  $0, %edx
  divq  %rcx
  imulq $16, %rax, %rax
  subq  %rax, %rsp
  movq  %rsp, %rax
  addq  $3, %rax
  shrq  $2, %rax
  salq  $2, %rax
  movq  %rax, -24(%rbp)
  movl  $0, %eax
  call  make_leaf
```

Массивы переменной длины


```
01. #include <stdlib.h>
02.
03. void make_leaf()
04. {
05.     // intentionally empty body
06. }
07.
08. int main()
09. {
10.     size_t n = 10000;
11.     int array[n];
12.     // don't ask why I'm here
13.     make_leaf();
14.     return 0;
15. }
```

```
01. main:
02. .LFB3:
03.   .cfi_startproc
04.   pushq %rbp
05.   .cfi_def_cfa_offset 16
06.   .cfi_offset 6, -16
07.   movq  %rsp, %rbp
08.   .cfi_def_cfa_register 6
09.   pushq %rbx
10.   subq  $40, %rsp
11.   .cfi_offset 3, -24
12.   movq  %rsp, %rax
13.   movq  %rax, %rbx
14.   movq  $10000, -40(%rbp)
15.   movq  -40(%rbp), %rax
16.   movq  %rax, %r8
17.   subq  $1, %r8
18.   movq  %r8, -32(%rbp)
19.   movq  %rax, %rsi
20.   movl  $0, %edi
21.   movq  %rax, %rdx
22.   movl  $0, %ecx
23.   salq  $2, %rax
24.   leaq  3(%rax), %rdx
25.   movl  $16, %eax
26.   subq  $1, %rax
27.   addq  %rdx, %rax
28.   movl  $16, %ecx
29.   movl  $0, %edx
30.   divq  %rcx
31.   imulq $16, %rax, %rax
32.   subq  %rax, %rsp
33.   movq  %rsp, %rax
34.   addq  $3, %rax
35.   shrq  $2, %rax
36.   salq  $2, %rax
37.   movq  %rax, -24(%rbp)
38.   movl  $0, %eax
39.   call  make_leaf
```

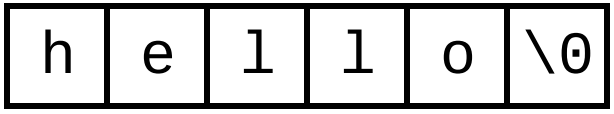
Строки в языке C

В языке C нет встроенного типа для хранения строк

Строки в языке C

В языке C нет встроенного типа для хранения строк

Строка — массив символов. Признак конца строки — байт со значением 0.



Ввод строк


```
char *fgets(char *s, int size, FILE *stream);
```

`fgets()` reads in at most one less than size characters from stream and stores them into the buffer pointed to by s. Reading stops after an `EOF` or a newline. If a newline is read, it is stored into the buffer. A terminating null byte (`'\0'`) is stored after the last character in the buffer.

Ввод строк


```
01. #include <stdio.h>
02.
03. enum {
04.     MaxNameLength = 40
05. };
06.
07. int main()
08. {
09.     char name[MaxNameLength];
10.
11.     printf("Enter name: ");
12.     fgets(name, MaxNameLength - 1, stdin);
13.
14.     printf("Hello, %s!\n", name);
15.
16.     return 0;
17. }
```

Длина строки и размер массива?


```
01. #include <stdio.h>
02. #include <string.h>
03.
04. int main()
05. {
06.     char name[4] = "Mary";
07.
08.     printf("sizeof(name) = %zu\n", sizeof(name));
09.     printf("strlen(name) = %zu\n", strlen(name));
10.
11.     return 0;
12. }
```

Длина строки и размер массива?


```
01. #include <stdio.h>
02. #include <string.h>
03.
04. int main()
05. {
06.     char name[4] = "Mary";
07.
08.     printf("sizeof(name) = %zu\n", sizeof(name));  // 4
09.     printf("strlen(name) = %zu\n", strlen(name));  // Undefined
10.
11.     return 0;
12. }
```

Длина строки и размер массива?


```
01. #include <stdio.h>
02. #include <string.h>
03.
04. int main()
05. {
06.     char name[] = "Mary";
07.
08.     printf("sizeof(name) = %zu\n", sizeof(name));
09.     printf("strlen(name) = %zu\n", strlen(name));
10.
11.     return 0;
12. }
```

Длина строки и размер массива?


```
01. #include <stdio.h>
02. #include <string.h>
03.
04. int main()
05. {
06.     char name[] = "Mary";
07.
08.     printf("sizeof(name) = %zu\n", sizeof(name));  // 5
09.     printf("strlen(name) = %zu\n", strlen(name));  // 4
10.
11.     return 0;
12. }
```

Длина строки и размер массива?


```
01. #include <stdio.h>
02. #include <string.h>
03.
04. int main()
05. {
06.     char name[] = {'M', 'a', 'r', 'y'};
07.
08.     printf("sizeof(name) = %zu\n", sizeof(name));
09.     printf("strlen(name) = %zu\n", strlen(name));
10.
11.     return 0;
12. }
```

Длина строки и размер массива?


```
01. #include <stdio.h>
02. #include <string.h>
03.
04. int main()
05. {
06.     char name[] = {'M', 'a', 'r', 'y'};
07.
08.     printf("sizeof(name) = %zu\n", sizeof(name));  // 4
09.     printf("strlen(name) = %zu\n", strlen(name));  // Undefined
10.
11.     return 0;
12. }
```

Задание 1

Реализовать алгоритм Bubble sort.

Пользователь вводит размер массива и значения элементов или только значения элементов при использовании массива фиксированного размера.

Приложение выводит отсортированный массив и количество выполненных итераций.

https://visualgo.net/sorting

Задание 2

Разработать приложение, проверяющее, является ли введенная пользователем строка палиндромом.

Примеры палиндромов:

- Hannah
- racecar
- I

Домашнее задание 1

Реализовать алгоритмы сортировки вставками и подсчетом.

Домашнее задание 2

Разработать приложение, проверяющее, является ли введенная пользователем фраза палиндромом.

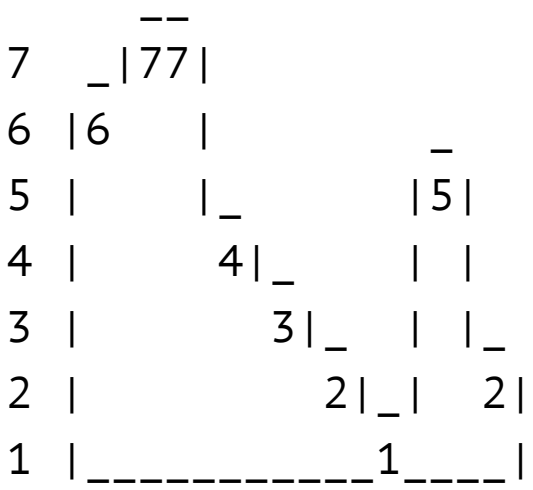
Примеры палиндромов:

- Never odd or even
- Madam in Eden, I'm Adam.

Если фраза не содержит ни одной буквы, вывести сообщение об ошибке.

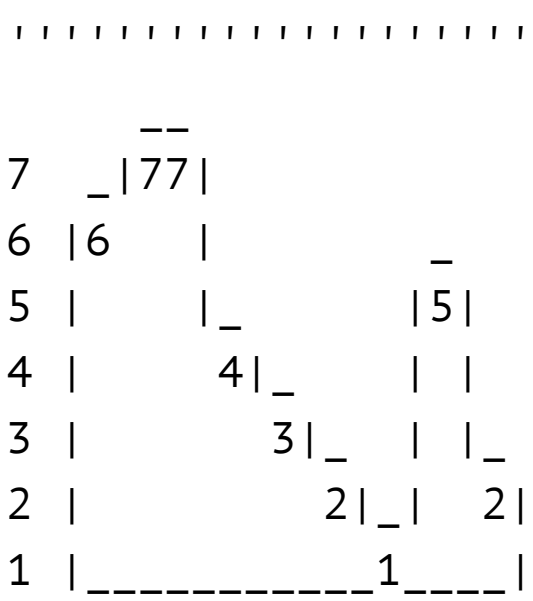
Домашнее задание 3*

Дан рельеф:



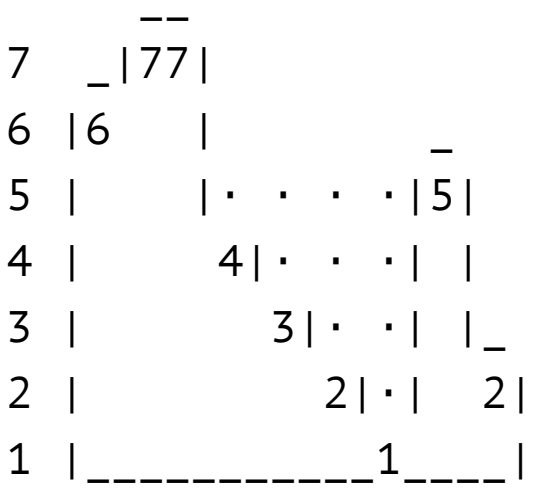
Домашнее задание 3*

Представим, что идет дождь:



Домашнее задание 3*

Сколько воды соберется в углублениях?



Ответ: 10

Домашнее задание 3*

Пользователь вводит значения одномерного массива, определяющие высоту «стен». Приложение вычисляет объем «воды», собравшейся между стенами.

- Единица объема воды — одна ячейка
- Слева и справа рельеф не ограничен. В приведенном примере вода с краев выплескивается.