

VLA, alloca

Variable Length Array

```
#include <stdio.h>

int main(void)
{
    int n;

    printf("n: ");
    scanf("%d", &n);

    int a[n];

    printf("n = %d, sizeof(a) = %d\n",
           n, (int) sizeof(a));
```

```
    for (int i = 0; i < n; i++)
        a[i] = i;

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

    return 0;
}
```

Variable Length Array

- Длина такого массива вычисляется во время выполнения программы, а не во время компиляции.
- Память под элементы массива выделяется на стеке.
- Массивы переменного размера нельзя инициализировать при определении.
- Массивы переменной длины могут быть многомерными.
- Адресная арифметика справедлива для массивов переменной длины.
- Массивы переменной длины облегчают описание заголовков функций, которые обрабатывают массивы.

alloca

```
#include <alloca.h>
```

```
void* alloca(size_t size);
```

Функция *alloca* выделяет область памяти, размером *size* байт, на стеке. Функция возвращает указатель на начало выделенной области. Эта область автоматически освобождается, когда функция, которая вызвала *alloca*, возвращает управления вызывающей стороне.

Если выделение вызывает переполнение стека, поведение программы не определено.

alloca

<pre>#include <alloca.h> #include <stdio.h> int main(void) { int n; printf("n: "); scanf("%d", &n); int *a = alloca(n * sizeof(int)); for (int i = 0; i < n; i++) a[i] = i;</pre>	<pre> for (int i = 0; i < n; i++) printf("%d ", a[i]); return 0; }</pre>
--	---

alloca

“+”

- Выделение происходит быстро.
- Выделенная область освобождается автоматически.

“_”

- Функция *нестандартная*.
- Серьезные ограничения по размеру области.

- ```
void foo(int size) {
 ...
 while(b) {
 char tmp[size];
 ...
 }
}
```

```
void foo(int size) {
 ...
 while(b) {
 char* tmp = alloca(size);
 ...
 }
}
```