**Динамическая память**

Для управления динамическим выделением памяти используется ряд функций, которые определены в заголовочном файле **stdlib.h**:

* **malloc()**. Имеет прототип

|  |  |
| --- | --- |
| 1 | void \*malloc(unsigned s); |

Выделяет память длиной в s байт и возвращает указатель на начало выделенной памяти. В случае неудачного выполнения возвращает NULL

**сalloc()**. Имеет прототип

|  |  |
| --- | --- |
| 1 | void \*calloc(unsigned n, unsigned m); |

Выделяет память для n элементов по m байт каждый и возвращает указатель на начало выделенной памяти. В случае неудачного выполнения возвращает **NULL**

**realloc()**. Имеет прототип

|  |  |
| --- | --- |
| 1 | void \*realloc(void \*bl, unsigned ns); |

Изменяет размер ранее выделенного блока памяти, на начало которого указывает указатель bl, до размера в ns байт. Если указатель bl имеет значение **NULL**, то есть память не выделялась, то действие функции аналогично действию **malloc**

**free()**. Имеет прототип

|  |  |
| --- | --- |
| 1 | void \*free(void \*bl); |

Освобождает ранее выделенный блок памяти, на начало которого указывает указатель bl.

Если мы не используем эту функцию, то динамическая память все равно освободится автоматически при завершении работы программы. Однако все же хорошей практикой является вызов функции **free()**, который позволяет как можно раньше освободить память.

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int \*mas; // указатель для блока памяти

int n,i; // число элементов массива

printf("Size of array=");// ввод числа элементов

scanf("%d", &n);

// выделяем память для массива

// функция malloc возвращает указатель типа void\*

// который автоматически преобразуется в тип int\*

/\* mas = malloc(n \* sizeof(int));\*/

mas = calloc(n, sizeof(int));

for(i=0;i<n; i++)// ввод вывод массива

{

mas[i]=rand()%100-50;

printf("%d \t", mas[i]); }

free(mas); // освобождаем память

return 0;

}

После выполнения всех действий память освобождается с помощью функции free():

Важно, что после выполнения этой функции мы уже не сможем использовать массив, например, вывести его значения на консоль:

**Двумерный массив**

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#define COL 5

#define ROW 10

void main() {

int \*\*p = NULL;

int i,j;

p = (int\*\*) malloc(ROW \* sizeof(int\*));

for (i = 0; i < ROW; i++)

p[i] = (int\*) malloc(COL \* sizeof(int));

for (i=0;i<ROW;i++)

{

for (j=0;j<COL;j++)

{

p[i][j]=rand()%100-50;

printf("%d\t",p[i][j]) ;

}

printf("%d\n");

}

for (i = 0; i < ROW; i++) {

free(p[i]);

}

free(p);

}

**Построение Треугольника Паскаля**

#include <conio.h>

#include <stdio.h>

#include <stdlib.h>

#define M 20

void main() {

int\*\* bin = NULL;

int i, j,n;

printf("Enter razmer: ");

scanf("%d", &n);

bin = (int\*\*) malloc(n \* sizeof(int\*));

for (i = 0; i < n; i++) {

bin[i] = (int\*) malloc((i + 1) \* sizeof(int));

}

bin[0][0] = 1;

for (i = 1; i < n; i++) {

bin[i][0] = bin[i][i] = 1;

for (j = i - 1; j > 0; j--) {

bin[i][j] = bin[i-1][j-1] + bin[i-1][j];

}

}

for (i = 0; i < n; i++) {

for (j = 0; j <= i; \

\

{

printf("%4d ", bin[i][j]);

}

free(bin[i]);

printf("\n");

}

free(bin);

getch();

}