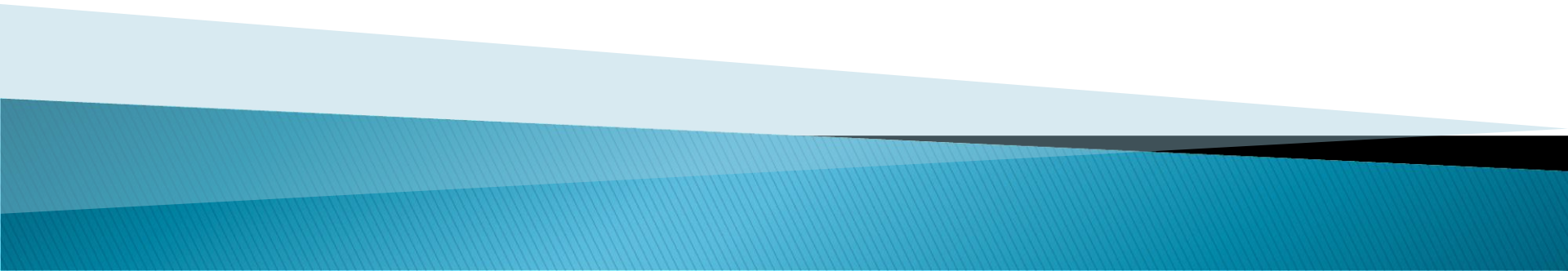


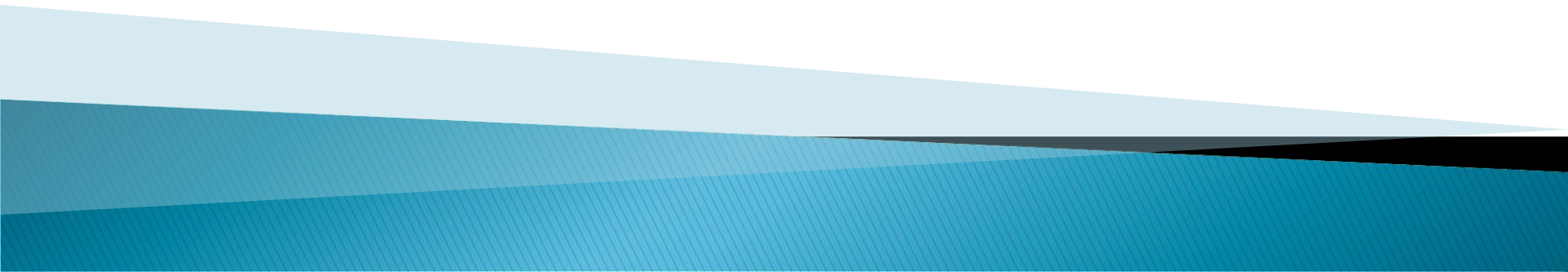
Саркисян Г.Ф.

ПРОГРАММИРОВАНИЕ

Часть I: Структурное программирование



РЕКУРСИВНЫЕ ФУНКЦИИ



РЕКУРСИВНЫЕ ФУНКЦИИ

Рекурсивная функция – это функция, которая вызывает сама себя – это **прямая рекурсия**, либо вызывает себя **косвенно** с помощью другой функции.

Если функция вызывает себя, то:

1. в стеке выделяется память под формальные параметры функции;
2. управление передается первому исполняемому оператору функции;

РЕКУРСИВНЫЕ ФУНКЦИИ

3. при повторном вызове данный процесс повторяется;

4. при завершении вызова функции соответствующая часть стека освобождается, и управление передается вызывающей функции, выполнение которой продолжается с точки, следующей за рекурсивным вызовом.

РЕКУРСИВНЫЕ ФУНКЦИИ

Рекурсивная функция умеет решать только простейшую часть задачи – так называемую **базовую задачу** (их может быть несколько).

Если функция вызывается для решения **базовой задачи**, то она просто возвращает результат.

Если функция вызывается для решения более сложной задачи, то задача делится на две части: на **базовую задачу** и **рекурсивный вызов** (шаг рекурсии).

РЕКУРСИВНЫЕ ФУНКЦИИ

Шаг рекурсии должен быть похож на исходную задачу, но по сравнению с ней должна быть более проще и включать в себя ключевое слово **return**.

Для завершения процесса рекурсии, каждый раз, как функция вызывает себя, должна формироваться последовательность все меньших и меньших задач в конце концов сходящаяся к базовой задаче.

РЕКУРСИВНЫЕ ФУНКЦИИ

Рекурсивные функции применяются:

1. для компактной реализации алгоритмов;
2. для работы со структурами данных.

Достоинством рекурсии является компактность записи.

Недостатками рекурсии – расход времени и памяти на повторные вызовы, а главная опасность переполнение стека.

РЕКУРСИВНЫЕ ФУНКЦИИ

Посчитать факториал числа.

$n! = n * (n-1) * (n-2) * \dots * 2 * 1$, причем $1! = 1$ и $0! = 1$.

```
int fact( int n){  
    // базовая задача  
    if (n == 1 || n == 0) return 1;  
  
    // шаг рекурсии  
    return n * fact(n-1);  
}
```


РЕКУРСИВНЫЕ ФУНКЦИИ

Данную задачу можно записать используя оператор “? :”

```
int fact ( int n) {  
    return ( n > 1) ? n * fact(n-1) : 1;  
}
```

РЕКУРСИВНЫЕ ФУНКЦИИ

Ряд Фибоначчи. x_n – ый элемент ряда Фибоначчи равняется сумме двух предыдущих элементов ряда, т.е. $x_n = x_{n-1} + x_{n-2}$.

Функция получает номер ряда и возвращает элемент под данным номером.

```
int fibonachi(int n) {  
    if( n == 1 || n == 0 ) return n;  
    return fibonachi(n-1)+fibonachi(n-2);  
}
```

РЕКУРСИВНЫЕ ФУНКЦИИ

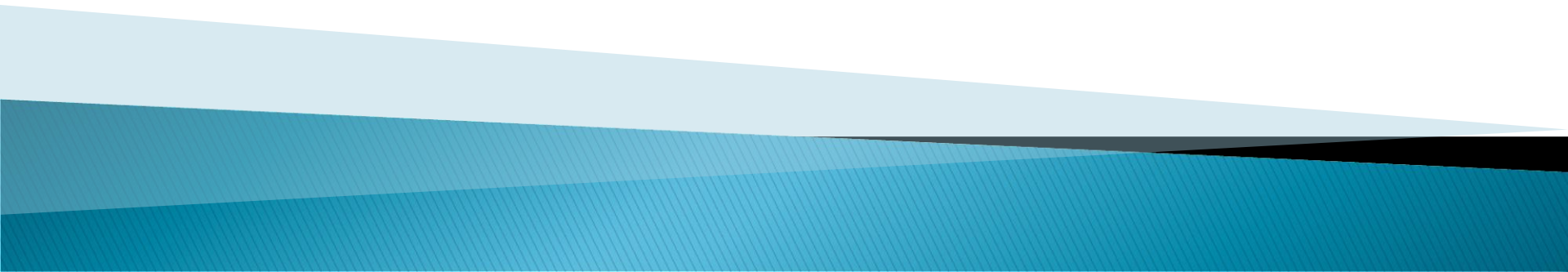
```
void sum_cifr( int, int* );  
void main(void) {  
    int n, sum=0;  
    printf("n = "); scanf("%d", &n);  
    sum_cifr( n, &sum);  
    printf("summa = %d\n", sum);  
}  
  
void sum_cifr(int n, int *p) {  
    if(n) { *p += n%10;    sum_cifr(n/10, p) }  
}
```

РЕКУРСИВНЫЕ ФУНКЦИИ

```
int sum_cifr(int) ;
void main(void) {
    int n;
    printf("n = ") ; scanf("%d", &n) ;
    printf("%d", sum_cifr(n)) ;
}

int sum_cifr(int n) {
    static int sum;
    if(n==0) return sum;
    sum += n%10;
    return sum_cifr(n/10) ;
}
```

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ



ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ

Для работы с генерацией случайных чисел необходимо подключить следующие библиотеки: `stdlib.h` и `time.h`

В библиотеке `stdlib.h` находятся прототипы функции:

```
int rand(void) ;
```

Функция возвращает числа в диапазоне от 0 до 32767.

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ

```
void srand( unsigned int) ;
```

Функция получает в качестве аргумента число в пределах от 0 до 65535, если **int** занимает 2 байта и от 0 до 4 294 967 295 если 4 байта.

Функция **srand** устанавливает начальное псевдослучайное число.

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ

Функцию `srand` – рекомендуется использовать следующим образом:

```
srand( time(NULL) );
```

где в качестве параметра используется вызов функции `time`, которая находится в библиотеке `time.h` и возвращает текущую дату в виде целого числа.

Получение случайного числа из интервала `[a , b]`

```
int n;           n = a + rand() % (b - a + 1);
```


СПАСИБО ЗА ВНИМАНИЕ!

ВОПРОСЫ ?

КЛАССЫ ПАМЯТИ

Автор: Саркисян Гаяне Феликсовна