

**Санкт-Петербургский государственный университет  
телекоммуникаций им. проф. М.А. Бонч-Бруевича**

**Подготовка к зачету**

**СПб ГУТ)))**

## ПЕРЕЧЕНЬ ВОПРОСОВ, ВЫНОСИМЫХ НА ЗАЧЕТ

1. Назначение и организация модулей в языке Си
2. Принцип одной задачи
3. Недостатки при совмещении обработки с вводом или выводом.
4. Связь между массивами и указателями.
5. Использование указателей для возврата из функции результатов вычислений.
6. Рекурсии. Достоинства и недостатки.
7. Недействительный указатель.
8. Ошибки при работе с динамической памятью. Висячий указатель.
9. Ошибки при работе с динамической памятью. Утечка памяти.
10. Идиома `*ptr++`.
11. Организация строк в языке Си. Общие сведения.
12. Ввод – вывод строк. Функции, используемые для ввода строк. Особенности ввода строк с помощью функций `scanf()` и `gets()`.

3. Строковая переменная в языке Си. Использование строк в качестве параметров функций.
4. Совместный ввод строк и чисел.
5. Обработка двумерного массива по строкам и столбцам.
6. Структуры в языке Си. Объявление структуры, ее инициализация.
7. Структуры в языке Си. Операции со структурами.
8. Структуры в языке Си. Средства обращения к полям структуры.
9. Указатель на структуру и его использование при обращении к полям структуры.
0. Использование структур в функциях.
1. Массивы структур.
2. Указатели на функцию. Разновидности указателей на функцию.
3. Использование указателей на функции в качестве параметров функции.
4. Работа с внешними устройствами. Общие сведения.
5. Файловый ввод – вывод. Форматированный файловый ввод – вывод.
6. Файловый ввод – вывод. Строковый файловый ввод – вывод.

# 1. Назначение и организация модулей в языке Си

```
7
8 #ifndef CAR_H
9 #define CAR_H
10
11 #ifdef __cplusplus
12 extern "C" {
13 #endif
14
15     struct Car{
16         char mark[20];
17         char color[20];
18         int age;
19     };
20
21 #ifdef __cplusplus
22 }
23 #endif
24
25 #endif /* CAR_H */
```

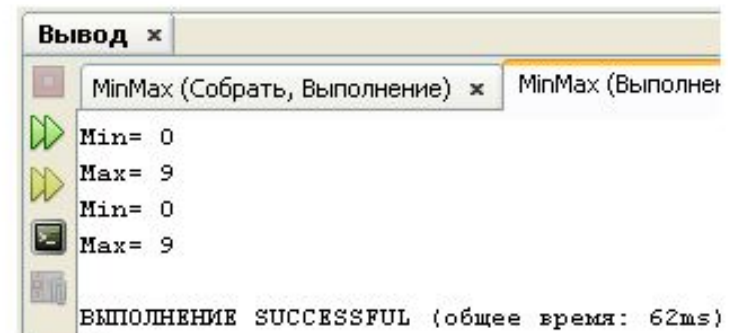
```
1 #include <cstdlib>
2 #include<iostream>
3 #include<string.h>
4 #include"Car.h"
5 void out_park(struct Car MyPark[], int n);
6 void inp_park(struct Car MyPark[], int n);
7 void edit_park(struct Car MyPark[]);
8 void calc_park(struct Car MyPark[],int i);
9 using namespace std;
10
11 int main(int argc, char** argv) {
12
```

## 2. Принцип одной задачи

## 3. Недостатки при совмещении обработки с вводом или выводом

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4 int Min (int arr[],int n)
5 {int i, m=arr[0];
6 for(i=1; i<n; i++)
7 if(arr[i]<m) m=arr[i];
8 return m;
9 // Плохое решение cout<<m;
10 int Max (int arr[],int n)
11 {int i, m=arr[0];
12 for(i=1; i<n; i++)
13 if(arr[i]>m) m=arr[i];
14 return m;
15 }
16 void MinMax (int arr[],int n, int &min, int &max)
17 {int i;
18 min=max=arr[0];
19 for(i=1; i<n; i++)
20 {
21 if(arr[i]>max) max=arr[i];
22 if(arr[i]<min) min=arr[i];
23 }
24 }
```

```
25 int main(int argc, char** argv) {
26 int arr[]={5,4,3,7,8,1,0,9,2,6};
27 int min, max;
28 cout<<"Min= " <<Min(arr,10) <<endl;
29 cout<<"Max= " <<Max(arr,10) <<endl;
30 MinMax(arr,10,min,max);
31 cout<<"Min= " <<min <<endl;
32 cout<<"Max= " <<max <<endl;
33 return 0;
34 }
```



```
Вывод x
MinMax (Собрать, Выполнение) x MinMax (Выполне
Min= 0
Max= 9
Min= 0
Max= 9
ВЫПОЛНЕНИЕ SUCCESSFUL (общее время: 62ms)
```

## 4. Связь между массивами и указателями

Имя массива является указателем на нулевой элемент массива. Далее элементы массива располагаются в памяти друг за другом. Для доступа к элементам массива существует два различных способа. Первый способ связан с использованием обычных индексных выражений в квадратных скобках, например, `array[16]=3` или `array[i+2]=7`. При таком способе доступа записываются два выражения, причем второе выражение заключается в квадратные скобки. Одно из этих выражений должно быть указателем, а второе - выражением целого типа. Вторым способом доступа к элементам массива связан с использованием адресных выражений и операции разадресации в форме `*(array+16)=3` или `*(array+i+2)=7`. При реализации на компьютере первый способ приводится ко второму, т. е. индексное выражение преобразуется к адресному. Для приведенного примера `array[16]` преобразуются в `*(array+16)`.

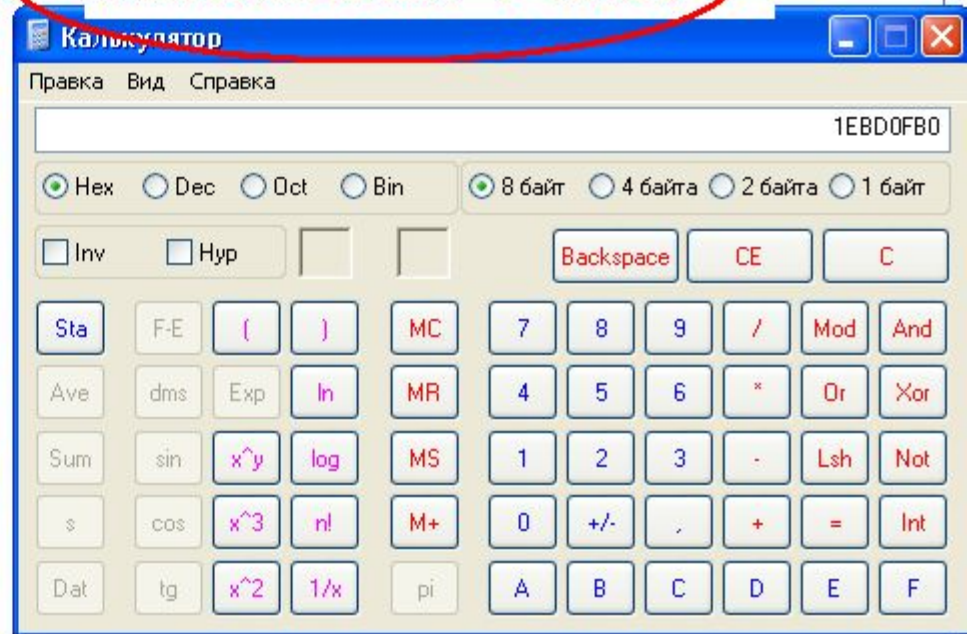
#### 4. Связь между массивами и указателями

```
double arr[]={0.1,1.1,2.1,3.1,4.1,5.1,6.1,7.1,8.1,9.1};
```

- `for(i=0;i<10;i++)`
- `{`
- `cout<<" arr["<<i<<"]="<<arr[i]<<" addr="<<&arr[i]<<endl;`
- `}`

- `arr[0]=0.1`    `addr=0x1ebd0fa8`
- `arr[1]=1.1`    `addr=0x1ebd0fb0`
- `arr[2]=2.1`    `addr=0x1ebd0fb8`
- `arr[3]=3.1`    `addr=0x1ebd0fc0`
- `arr[4]=4.1`    `addr=0x1ebd0fc8`
- `arr[5]=5.1`    `addr=0x1ebd0fd0`
- `arr[6]=6.1`    `addr=0x1ebd0fd8`
- `arr[7]=7.1`    `addr=0x1ebd0fe0`
- `arr[8]=8.1`    `addr=0x1ebd0fe8`
- `arr[9]=9.1`    `addr=0x1ebd0ff0`

переменная типа double занимает в памяти 8 байт.  $1\text{ebd0fa8} + 8 = 1\text{ebd0fb0}$



## 5. Использование указателей для возврата из функции результатов вычислений.

```
15 void MinMax (int arr[],int n, int &min, int &max)
16 {int i;
17 min=max=arr[0];
18 for(i=1; i<n; i++)
19 {
20     if(arr[i]>max) max=arr[i];
21     if(arr[i]<min) min=arr[i];
22 }
23 }
24 void PMinMax (int arr[],int n, int *min, int *max
25 {int i;
26 *min=*max=arr[0];
27 for(i=1; i<n; i++)
28 {
29     if(arr[i]>*max) *max=arr[i];
30     if(arr[i]<*min) *min=arr[i];
31 }
32 }
```

```
33 int main(int argc, char** argv) {
34     int arr[]={5,4,3,7,8,1,0,9,2,6};
35     int min, max;
36     cout<<"Min= " <<Min(arr,10)<<endl;
37     cout<<"Max= " <<Max(arr,10)<<endl;
38     MinMax(arr,10,min,max);
39     cout<<"Min= " <<min<<endl;
40     cout<<"Max= " <<max<<endl;
41     PMinMax(arr,10, &min, &max);
42     cout<<"Min= " <<min<<endl;
43     cout<<"Max= " <<max<<endl;
44     return 0;
45 }
```

Вывод ×

MinMax (Собрать, Выполнение) ×	MinMax (Выполнение) ×
Min= 0	
Max= 9	
Min= 0	
Max= 9	
Min= 0	
Max= 9	



6. Рекурсии. Достоинства и недостатки.

Смотри файлы С++ Рекурсия.ppt и Статья про рекурсию.pdf

7. Недействительный указатель.

Смотри файл ч\_2\_Практикум.doc стр. 21

8. Ошибки при работе с динамической памятью. Висячий указатель  
Смотри файл ч\_2\_Практикум.doc стр. 34

.

9. Ошибки при работе с динамической памятью. Утечка памяти.  
Смотри файл ч\_2\_Практикум.doc стр. 34

.

10. Идиома `*ptr++`.

Смотри файл `ч_2_Практикум.doc` стр. 85

.

11. Организация строк в языке Си. Общие сведения.
12. Ввод – вывод строк. Функции, используемые для ввода строк.  
Особенности ввода строк с помощью функций `scanf()` и `gets()`.
13. Строковая переменная в языке Си. Использование строк в качестве параметров функций.
14. Совместный ввод строк и чисел.

Смотри папку Лек 1 (строки)

.

11. Обработка двумерного массива по строкам и столбцам.

Смотри файл ч\_2\_Практикум.doc стр. 81

.

16. Структуры в языке Си. Объявление структуры, ее инициализация.
17. Структуры в языке Си. Операции со структурами.
18. Структуры в языке Си. Средства обращения к полям структуры.
19. Указатель на структуру и его использование при обращении к полям структуры.
20. Использование структур в функциях.
21. Массивы структур.

Смотри Смотри папку Лек 3 (структ) и файл ч\_2\_Практикум.doc стр. 45

.