

# Домашнее задание № 4

*Study-Inf/1 курс/ПИ/ Информатика и программирование*

*Study-Inf/1 курс/БИ/ Информатика*

*Домашние задания и самостоятельная работа*

*Часть 2. Сдается бумажный вариант*

# КОНТРОЛЬНАЯ РАБОТА

## **Тема «Циклы в языке Си»**

гр. 405, 425-2 – 19 октября

гр. 425-1 – 21 октября

## **Примерный вариант задания**

Не используя массивы, ввести с клавиатуры натуральное число  $n$ , вывести на экран последовательность  $2*i + 3$ , для  $i$  от 1 до  $n$

# Динамические массивы

Описание:

```
int *x;
```

```
x = (int*) malloc(sizeof(int)*20); ...
```

- *обработка массивов произвольной размерности*
- *обработка массивов, размер которых превышает размер области данных*
- *передача массива параметром функции*

# Основные ошибки при работе с динамическими массивами

- Не выделяется память под массив
- Не освобождается память из под массива
- Обращение к **несуществующим** элементам

```
float *x;  
int n = 20;  
for (i=1;i<n;i++)  
    x[i]=2*i+1;
```

```
int *y =  
(int*)malloc(sizeof(int)*20) ;  
for (i=1;i<=n;i++)  
    y[i]=2*i+1;
```

## *Инициализация массивов*

Ввод элементов массива с клавиатуры:

```
int main(int argc, char *argv[])  
{  
    int *x;  
    int n;  
    printf("Введите размер массива: ");
```

```
scanf("%d",&n);  
x = (int*) malloc(sizeof(int)*n);  
int i;  
for(i=0;i<n;i++){  
    printf("x[%d] -> ",i);  
    scanf("%d",&x[i]);  
}  
printf("\nВведен массив: \n");  
for(i=0;i<n;i++){  
    printf("%5d",x[i]);  
}  
free(x);  
}
```

# Вычисление элементов массива по формуле

```
int main(int argc, char *argv[])  
{  
    int *x;  
    int n;  
    printf("Введите размер массива: ");  
    scanf("%d",&n);
```

```
x = (int*)malloc(sizeof(int)*n);  
int i;  
for(i=0;i<n;i++){  
    x[i] = i+2*i;  
    printf("%d ",x[i]);  
}  
    free(x);  
}
```



# Использование датчика случайных чисел

```
...  
for( i=0;i<n;i++)  
{  
    x[i] = rand()%20;  
    printf("%d ",x[i]);  
}...
```

Положительные  
числа  
в интервале [0;19]

```
...  
for(i=0;i<n;i++){  
    x[i] = rand()%31-rand()%31;  
    printf("%d ",x[i]);  
}...
```

Целые числа  
в интервале [-30;30]

# Получение вещественных значений

```
float *x;  
  
...  
x = (float*)malloc(sizeof(float)*n);  
for(int i=0;i<n;i++){  
x[i] = rand()%101/(rand()%31+1.);  
printf("%7.2f ",x[i]);  
}  
  
...
```

# Поиск значений в массиве

## Поиск по заданному значению



## Поиск индексов всех элементов

```
#include <time.h>
int main()
{
    printf(«Введите размерность массива: ");
    int n,i;
    srand(time(NULL));
    scanf("%d",&n);
    int *x = (int*)malloc(sizeof(int)*n);
```

```
for(i=0;i<n;i++)
{ x[i]=rand()%100-rand()%50;
  printf("%5d",x[i]);
}
printf("\n");
int flag = 0;
int k;
printf(«Введите значение для поиска:");
scanf("%d",&k);
for(i=0;i<n;i++)
  if (x[i]==k) { printf("x[%d]==%d\n",i,k);
                flag = 1;
              }
```

```
if (!flag)  
printf("В массиве нет элементов со значением  
    %d: ",k);  
free(x);  
}
```

## Нахождение первого заданного значения:

```
...  
printf(«Введите значение для поиска:»);  
scanf("%d",&k);  
for(i=0;i<n;i++)  
    if (x[i]==k) break;  
if (i!=n)  
    printf("Первый встреченный элемент со  
        значением %d расположен в позиции %d",  
        k,i);  
else  
    printf("В массиве нет элементов со значением  
        %d", k);
```

...

# Нахождение первого и второго вхождения заданного элемента

```
int k;  
printf("Enter k:");  
scanf("%d",&k);  
int flag = 0;  
int i1,i2;  
i=0;  
while(flag!=2&& i<n){  
    if (x[i]==k) { flag++;  
                  if (flag==1) i1=i;  
                  else i2=i; }  
    i++;  
}
```



```
i++;
```

```
}
```

```
if (flag==2) printf("1-й - %d, 2 й- %d ",i1,i2);
```

```
else
```

```
if(flag==1) printf("Один элемент - %d",i1);
```

```
else printf("Нет таких элементов");
```

```
}
```

## Некорректная реализация поиска номера последнего элемента массива с заданным значением

```
int h=-1;  
for(i=0;i<n;i++)  
    if (x[i]==k) h=i;  
if (h==-1) printf("В массиве нет таких  
элементов");  
else printf("Последний элемент находится  
в позиции %d",h);
```

Дан массив  $X[n]$ . Найти количество четных элементов массива

```
...  
int ch = 0;  
for(i=0;i<n;i++)  
    if (x[i]%2==0) ch++;  
if (ch!=0)  
    printf(" В массиве %d четных элементов ", ch);  
else  
    printf(" В массиве нет элементов с четными значениями");  
...
```

# Поиск экстремальных значений

## Алгоритм поиска минимального значения

<b>x[0]</b>	<b>x[1]</b>	<b>x[2]</b>	<b>x[3]</b>	<b>x[4]</b>
<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>

<b>2</b>	<b>1</b>	<b>7</b>	<b>0</b>	<b>3</b>
----------	----------	----------	----------	----------

<b>2</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>
<i>min</i>	<i>min</i>	<i>min</i>	<i>min</i>	<i>min</i>

```
if  
(min > x[i])  
  min = x[i];
```

...

```
min=x[0]; imin=0;
```

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

```
    if (min>x[i]) {min = x[i]; imin=i;} 
```

```
printf("Минимальное значение %d",min)
```

...

Поиск индекса (ов)

экстремального  
элемента

Поиск экстремального  
элемента  
индекса

Поиск количества  
экстремальных  
элементов

## Поиск индекса минимального элемента

```
int k=0;  
for(i=1;i<n;i++)  
    if (x[k]>x[i]) k = i;
```

Дан массив размерности  $n$  элементов.  
Найти значения трех самых маленьких значений массива.

25

30

10  
0

**$\text{min1} \leq \text{min2} \leq \text{min3}$**

15

45

45

2  
5  
3  
0  
4  
5  
1  
5  
15

```
#define MAXINT    0x7FFF
int main(int argc, char *argv[])
{
    printf("Введите размерность массива: ");
    int n,i;
    srand(time(NULL));
    scanf("%d",&n);
    int *x = (int*)malloc(sizeof(int)*n);
```

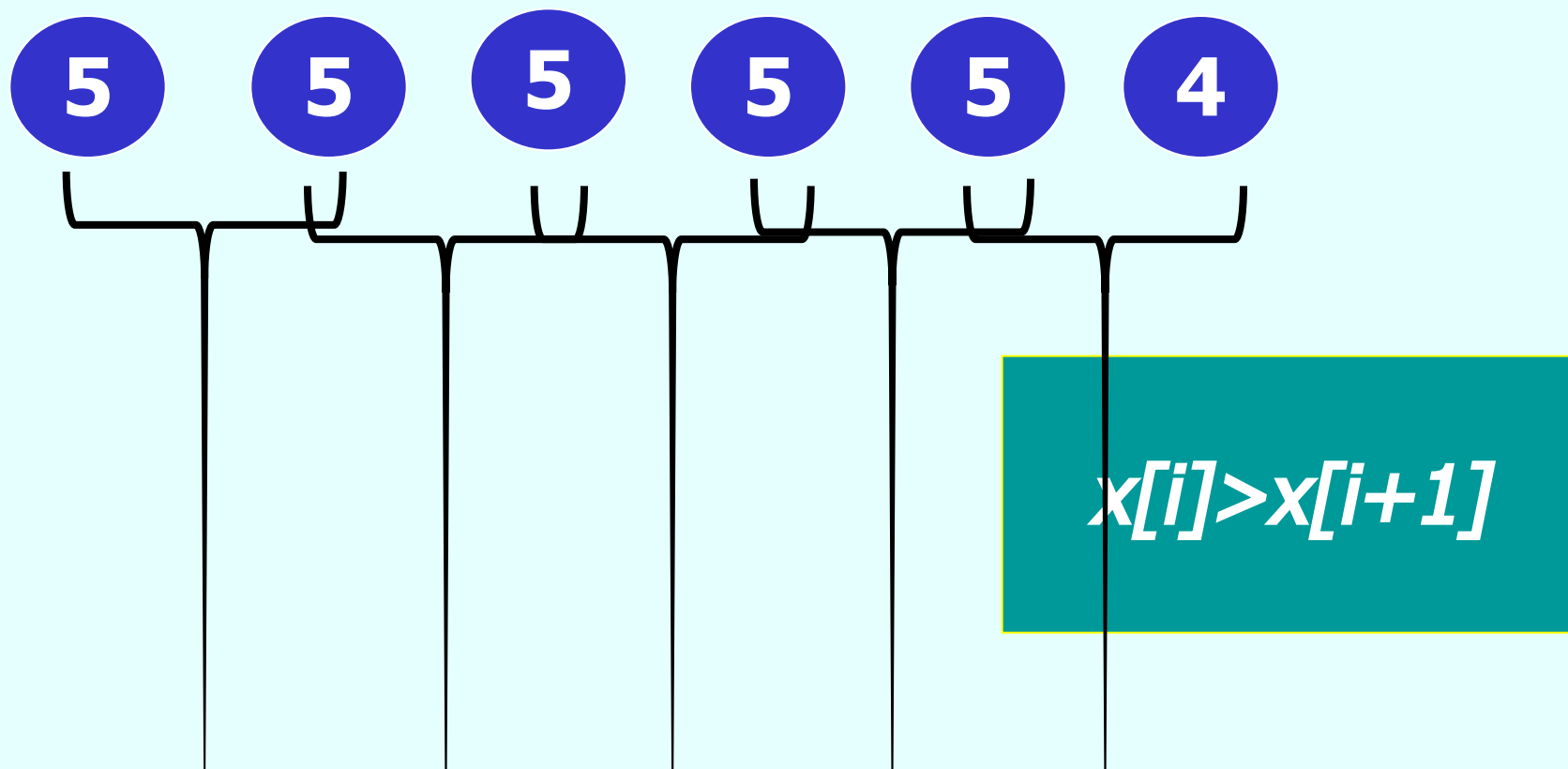


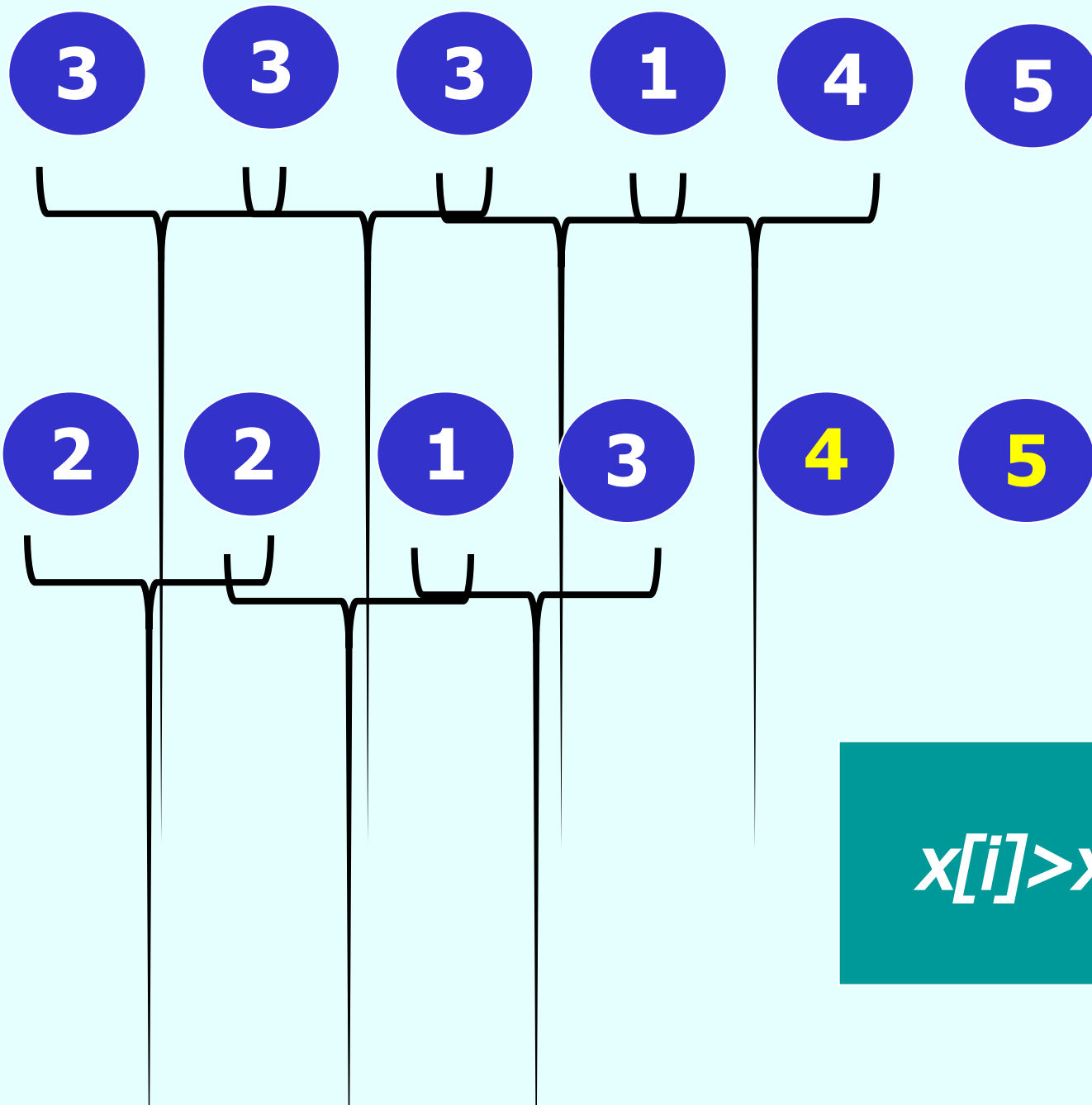
```
for(i=0;i<n;i++)  
    { x[i]=rand()%100;  
      printf("%5d",x[i]);  
    }  
printf("\n");  
int min1=MAXINT;  
int min2=MAXINT;  
int min3=MAXINT;  
for (i=0;i<n;i++)
```

```
if (x[i]<min1) {  
    min3 = min2;  
    min2 = min1;  
    min1 = x[i]; }  
    else if (x[i]<min2) {min3 = min2;  
        min2 = x[i];}  
    else if (x[i]<min3) min3 = x[i];  
printf("Три минимальных элемента %d %d  
      %d", min1,min2,min3);  
free(x);  
}
```

# Сортировка массивов

Сортировка обменом (пузырьковая сортировка)





$x[i] > x[i+1]$

...

```
for (i=0; i<n-1; i++){  
    flag = 0;  
    for (j=0; j<n-i-1; j++)  
        if (x[j]>x[j+1]) {  
            flag = 1;  
            temp = x[j];  
            x[j]=x[j+1];  
            x[j+1]=temp; }  
    if (flag==0) break;
```

...

# Сортировка выбором

5 3 2 0 1 4

5 3 2 0 0

$i=0$

*Поиск  
минимального  
элемента на  
текущую  
позицию*

0 3 2 5 1 4

$i=1$

3 2 2 1 1

0 1 2 5 3 4

$i=2$

2 2 2

0 1 2 5 3 4

$i=3$

5 3 3

0

1

2

3

5

4

$i=4$

5

4

...

```
for(int i=0;i<n-1;i++)
```

```
{ int k = i;
```

```
  for (int j=i+1;j<n;j++)
```

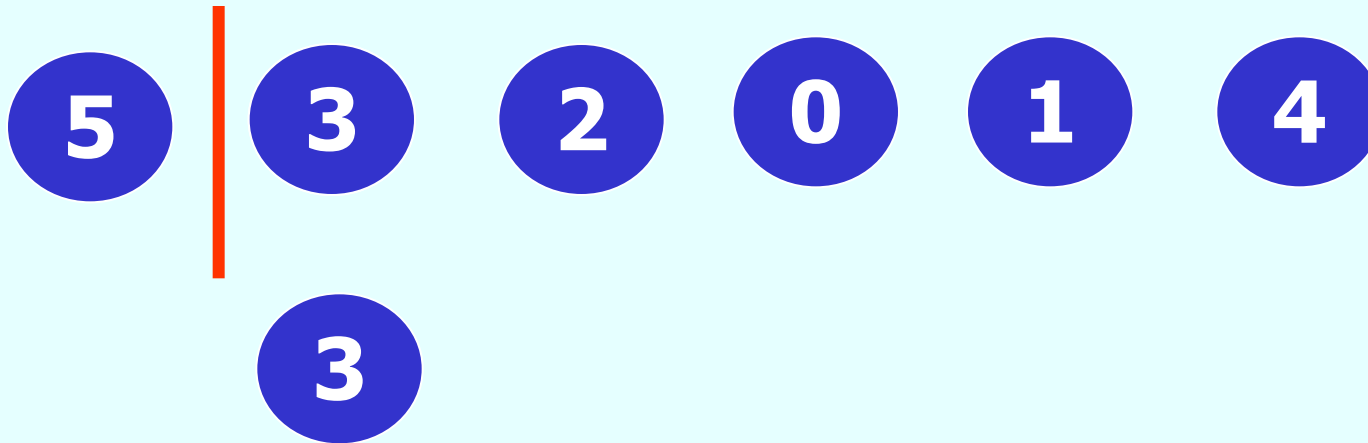
```
    if (x[k]>x[j]) k=j;
```

```
    if(k!=i) { int temp=x[i]; x[i]=x[k] ;  
x[k]=temp;}
```

```
} ...
```



# Сортировка вставками



$vs = x[1]$

$j = -1$

Пока  $vs < x[j]$  и  $j >= 0$

3

5



2

0

1

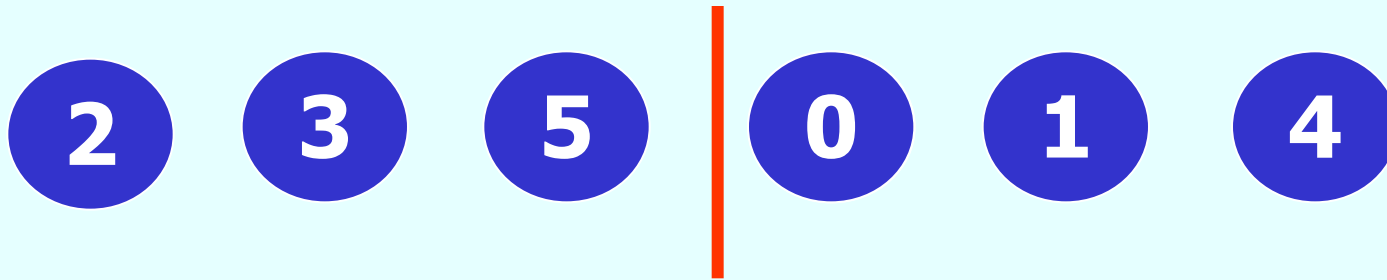
4

$vs = x[2]$

2

$j = -1$

Пока  $vs < x[j]$  и  
 $j >= 0$



0

$vs = x[3]$

$j = -1$

Пока  $vs < x[j]$  и  
 $j >= 0$

0

2

3

5

1

4

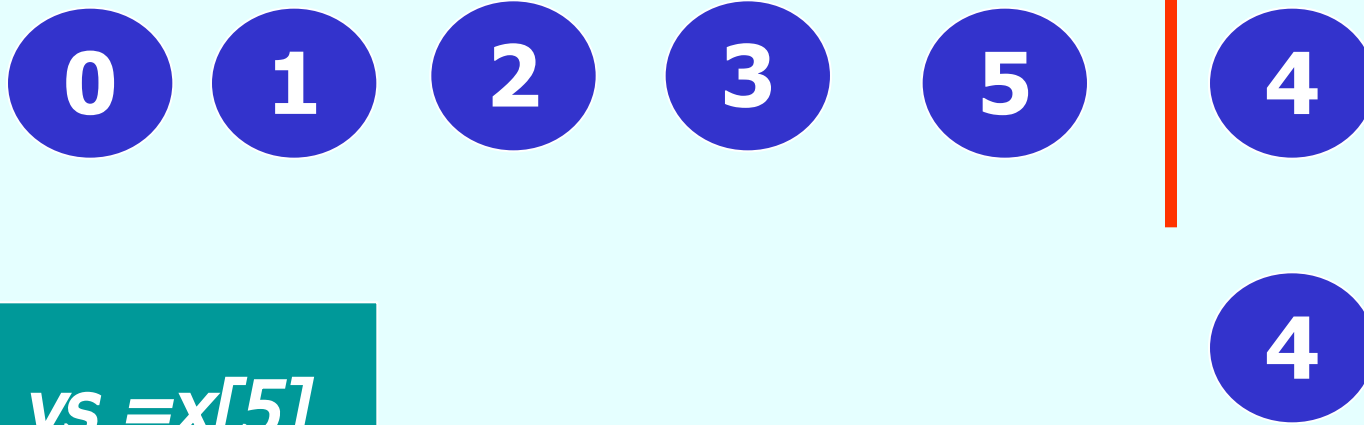


1

$vs = x[4]$

$j=0$

Пока  $vs < x[j]$  и  
 $j >= 0$



$vs = x[5]$

$j=3$

Пока  $vs < x[j]$  и  
 $j >= 0$