



Лекция 2 (продолжение)

**Массивы, описание,
передача в функцию в
качестве параметров**



Пример со строками (память!!!)



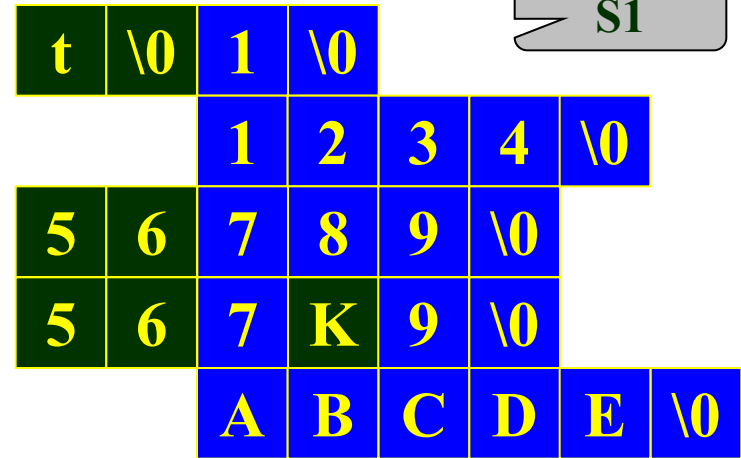
S2

S1

```

#include <stdio.h>
void main()
{char s1[]="1", s2[]="t";
printf("s1 "); scanf("%s", s1);
printf("s2 "); scanf("%s", s2);
printf("1____s1= %s\n", s1);
printf("2____s2= %s\n", s2);
s2[3]='K';
printf("3____s1= %s\n", s1);
printf("4____s2= %s\n", s2);
printf("s1 "); scanf("%s", s1);
printf("5____s2= %s\n", s2);}

```



1____s1= 789

2____s2= 56789

3____s1= 7K9

4____s2= 567K9

s1 ABCDE

5____s2= 56ABCDE



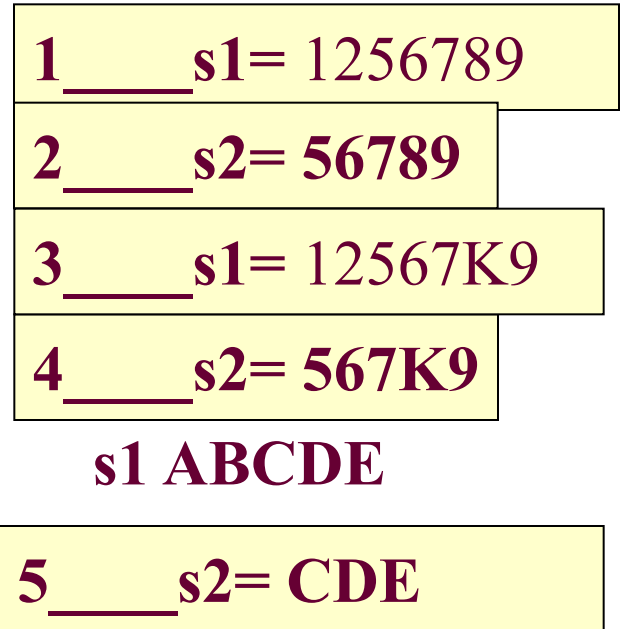
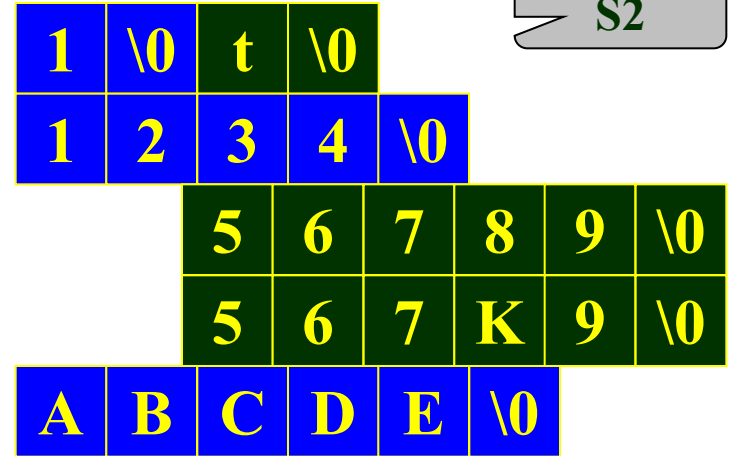
Строки - указатели (память!!!)



S1

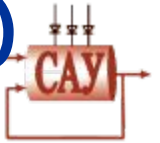
S2

```
#include <stdio.h>
void main()
{char *s1="1", *s2[]="t";
printf("s1 "); scanf("%s", s1);
printf("s2 "); scanf("%s", s2);
printf("1____s1= %s\n", s1);
printf("2____s2= %s\n", s2);
s2[3]='K';
printf("3____s1= %s\n", s1);
printf("4____s2= %s\n", s2);
printf("s1 "); scanf("%s", s1);
printf("5____s2= %s\n", s2);}
```





Динамические массивы (ввод массива)



```
int GetArray(int *p)
{
    int i, N;
    printf("n=");
    scanf("%d", &N);
    realloc(p, N);
    if (p == NULL) return(0);
    printf("input %d elements =", N);
    for (i=0; i<N; i++)
        scanf("%d", &p[i]);
    fflush(stdin);
    return(N);
}
```

На входе - адрес
первого элемента

На выходе число
элементов

Перевыделение памяти
на N элементов



Ввод одномерного массива



```
#include <stdio.h>
#include <alloc.h>
void main(void)
{ int *p=(int *) malloc(1);
  int n=0, i;
  fflush(stdin);
  n=GetArray(p);
  if (p!=NULL)
    for (i=0; i<n; i++)printf("%d ", p[i]);
  else printf("_____");
  printf("\n");
  free(p);
  getch();
}
```

выделение памяти на 1 элемент

Ввод массива с увеличением памяти



Три варианта расчета суммы



```
int sum0(int A[],int n)
{ int s,i;
for (i=s=0; i<n; i++)
    s += A[i];
return(s);}
```

```
int sum1(int *p, int n)
{ int s,i;
for (i=s=0; i<n; i++)
    s += p[i];
return(s);}
```

```
int sum2 ( int *p, int n)
{ int s;
for (s=0; n >0; n--)
s += *p++;
return(s); }
```

Выбор элемента по номеру

Выбор элемента по текущему адресу и перемещение указателя на следующий элемент



Проверка вычисления суммы



```
void main(void)
{ int*p=(int *) malloc(1);
  int n, i;
n=GetArray(p);
printf("s1=%d \n", sum1(p,n));
printf("s2=%d \n", sum2(p,n));
printf("s0=%d \n", sum0(p,n));
free(p);
  fflush(stdin) ;
  getchar();
}
```



Ввод матрицы из файла (1)



```
#define LINES 31
#define COLUMNS 79
void Input_Matrix(int *n, int *m,
    float MATR[LINES][COLUMNS],char *file_inp)
{int i , j;
FILE *f1; f1=fopen(file_inp,"r");
fscanf(f1,"%i %i\n", n,m);
    for(i=0; i < *n; i++)
        for(j=0; j < *m; j++)
            fscanf(f1,"%f",&MATR[i][j]);
fclose(f1);}
```

Вызов

```
int m=0, n=0;
float A[LINES][COLUMNS];
Input_Matrix(&n , &m, A,"M.dat");
```




Ввод матрицы из файла (2)



```
void Input_Matrix (int *n, int *m,  
    float MATR [ LINES ] [ COLUMNS ] ,char *file_inp)  
void Input_Matrix (int *n, int *m,  
    float MATR [ ] [ COLUMNS ] , char *file_inp)
```

```
void Input_Matrix (int *n, int *m,  
    float *MATR ,char *file_inp)  
{int i , j;  
FILE *f1;f1=fopen(file_inp,"r");  
fscanf(f1,"%i %i\n", n,m);  
    for(i=0; i < *n; i++)  
        for(j=0; j < *m; j++)  
            fscanf(f1,"%f", &MATR [ i * (*m) + j ]  
    );  
fclose(f1);  
}
```

Пересчет индекса

