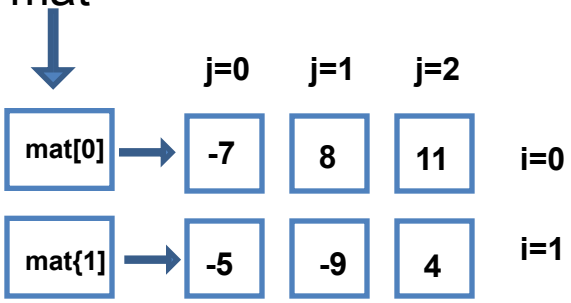
Приклад роботи з матрицями: створення матриці

```
int main()
{
printf("Введите количество строк и количество  столбцов матрицы  ");
scanf("%d%d", &n, &m); //n - рядків, m - стовців
int **mat=(int**) malloc(n* sizeof (int*)); //виділення пам'яті
for (int i=0; i<n; i++)
       mat[i]=(int *) malloc(m*sizeof (int));
InitScan (mat, n, m);
return 0;
}
```

```
void InitScan (int **mat, int n, int m)
{
  for (int i=0; i<n; i++ )
  {   printf("Введіть %d рядок\n",i);
      for (int j=0; j<m; j++ )
            {printf ("mat[%d][%d]= ", i, j);
            scanf("%d", &mat[i][j]);}
  }
}
```

n = 2 m=3

mat



Приклад роботи з матрицями: заповнення випадковими числами та друк матриці

Функція заповнення матриці випадковими числами

```
void InitRand (int **mat, int n, int m)
{ int range_max=-5, range_min=5; //задання меж розкиду параметру у рандомі
   srand(time(NULL));
   for (int i=0; i<n; i++ )
        for (int j=0; j<m; j++ )
       mat[i][j]= (double)rand() / (RAND_MAX + 1) * (range_max - range_min)+
range_min;
}
```

Функція роздруку значень матриці

```
void Print (int **mat, int n, int m)
{
      for (int i=0; i<n; i++ )
      {
            for (int j=0; j<m; j++ )
                  printf("a[%d][%d]=%d\t", i, j, mat[i][j]);
            printf("\n");
      }
}
```

Приклад роботи з матрицями: створення масиву характеристик та сортування стовпців матриць

```
void SortSumColumn (int **matr, int  n, int m)
{
  int *sumCol  = (int *) calloc (m , sizeof(int));
  //виділення памяті та обнулення масиву характеристик
   for (j = 0; j < m; j++)
      for (i = 0; i < n; i++)
                  sumCol[j] += matr[i][j];
            //знаходження суми у стовчику
  SortColumn(matr, sumCol, n, m); //виклик процедури сортування
   free(sumCol);//звільнення памяті
 }
```

Матриця 4 на 3

	j=0	j=1	j=2
i=0	4	3	1
i=1	7	9	1
i=2	1	2	2
i=3	3	7	3
	14	21	7

Масив характеристик

Аналогічно знаходиться сума у 1-му стовпчику

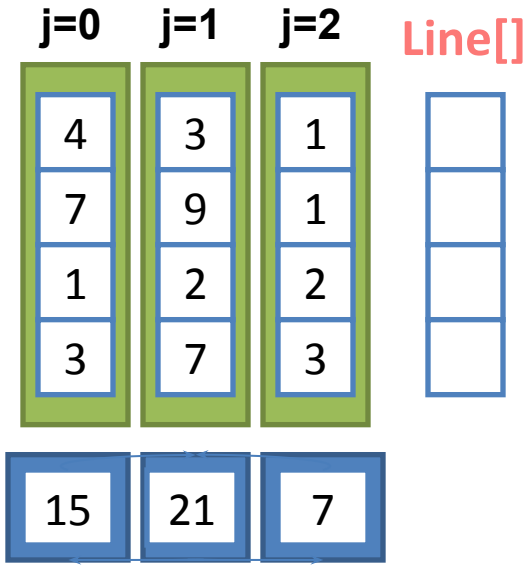
Аналогічно знаходиться сума у 2-му стовпчику

Виклик процедури сортування – наступний слайд

Приклад роботи з матрицями: створення масиву характеристик та сортування стовпців матриць

```
void SortColumn (int **matr, int *sumCol, int n, int m)
{
//сортування бульбашкою відповідно до зростання характеристик
int c, found;
int *Line = (int *) calloc (n, sizeof(int)); //створення стовпчику обміну
 do
 { found =0;
    for (int j = 0; j < m-1; j++)
    {   if (sumCol[j] > sumCol[j + 1])
    //якщо характеристика більша стовпчик переміщується вправо
          { c = sumCol[j]; sumCol[j]=sumCol[j + 1]; sumCol[j + 1]=c;
            //міняємо місцями характеристики
         for (int i = 0; i < n; i++)
         {  Line[i] = matr [i][j]; matr[i][j]=matr[i][j + 1]; matr[i][j + 1]=Line[i];
           //міняємо місцями j-й та j+1 стовпчик
            found=1;
         }
          }
    }
 } while(found !=0);
 free(Line);
}
```

Матриця 4 на 3

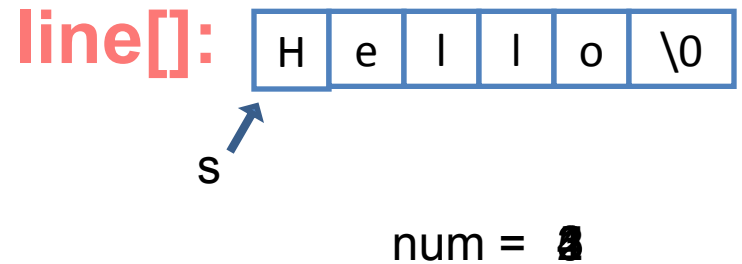


Масив характеристик

Приклад роботи з символьними рядками через покажчики:

Цикл for виконується доти, доки *s=‘\0’, що є спеціальним символом кінця рядка.

```
int main()
{
  int num=0;
  char line[100]; //статичне виділення пам’яті під масив із 100 символів
  gets(line); //ввід символів з клавіатури
  char *s; //змінна покажчик на char
  s=line; //s вказує на початок line
  for( ; *s; s++ ) //s проходить по line
     num++; //num рахує к-сть символіів
  printf("Kilkist simvoliv=%d", num);
  return 0;
}
```



На останньому кроці *s=‘\0’ , що є спеціальним символом кінця рядка.

Тому результат перевірки умови в for є true, що забезпечує вихід із циклу

Результат роботи програми:

```
hello
*s  = h, num= 1
*s  = e, num= 2
*s  = l, num= 3
*s  = l, num= 4
*s  = o, num= 5
Kilkist simvoliv=5
```

Приклад користувацьких функцій роботи з словами:
знаходження найдовшого слова

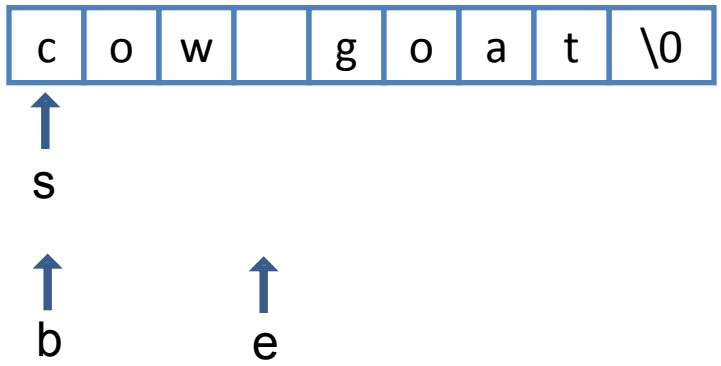
isalnum(int c) перевірка, чи є символ літерою або цифрою;

1 – повертає якщо символ є літерою або цифрою; 0 – у протилежному випадку

Функція повертає покажчик на початок слова

```
char *strwordb (char* s)
{  for( ; *s ; s++ ) //прохід по рядку
   if (isalnum(*s)) //якщо символ літера
        return s; //то повертаємо посилання
   return s;}
```

line[]:



Функція повертає покажчик на кінець слова

```
char *strworde (char* s)
{  for( ; *s ; s++ ) //прохід по рядку
    if (!isalnum(*s)) //якщо символ пробіл чи знак
        return s; //то повертаємо посилання
   return s;}
```

b – e = 3 4

max = 3 4

Функція повертає максимальну довжину слова

```
int MaxWord (char *s)
{ int max=0;
 char *b = strwordb(s),  char *e = NULL;
 for ( ;*b ;b = strwordb(e) ) {
    e=strworde(b);
    if( (e - b) > max) max = (e - b);}
return max;}
```

```
int  main()
{
 char line[] = "cow goat";
 printf("Najdovshe slovo = %d\n", MaxWord (line));
 return 0;
}
```