

## Пример с динамичен масив от низове създаден от текстови файл ( примерът може да се свали от [ТУК](#) )

Създава се динамично резервиран масив от указатели, всеки елемент от който се насочва към динамично резервиран масив от символи (ред прочетен от текстови файл). Всеки път когато масивът от указатели се изчерпи, се създава нов с удвоен размер.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define DELTA 5 /* начален размер на масива от указатели*/

/*функция за включване на ред прочетен в буфер */
char** include(char *bf,char** p,int *sizeAd,int* nm) {
    /******
    bf - указател към буфера, в който е реда за включване
    p - указател към масива от указатели към динамичните низове
    *sizeAD - размер на масива от указатели
    *nm - броя на включените низове
    функцията връща указател към новосъздадения масив от указатели
    *****/
    int i, ok =1; /* ok -флаг за проверка, че паметта е резервирана успешно */
    char **tmp;
    if(*nm<*sizeAd) { /*има място в създадения масив от указатели */
        /*-включване на съдържанието на реда в масива-- */
        if ((p[*nm]=(char*)malloc(strlen(bf)+1))==NULL) ok=0; /* Запазва се място за реда */
        else strcpy(p[*nm++],bf); /* и се копира от буфера */
        /*-----*/
    }
    else { /* масивът с указатели е запълнен */
        *sizeAd*=2; /*удвоява се размерът на масива от указатели */
        tmp=(char **)realloc(p,(*sizeAd)*sizeof(char*)); /* прави се нов масив от указатели */
        if(tmp!=NULL) { /* успешна операция - масивът от указатели е с удвоен размер */
            p=tmp;
            /*----включване на съдържанието на реда в масива-- */
            if ((p[*nm]=(char*)malloc(strlen(bf)+1))==NULL) ok=0; /* място за реда, при неуспех ok=0 */
            else strcpy(p[*nm++],bf); /* и се копира от буфера */
            /*-----*/
        }
        else { ok=0; } /* няма място за масив от указатели с удвоен размер */
    }
    if(!ok) {
        for( i=0;i<*nm;i++){
            free(p[i]); /* трябва да се освободи паметта за отделните редове */
        }
        free(p); /* освобождаване на масива от указатели */
        p=NULL;
    }
    return p;
}
```

```

int main(){
    char **ptr=NULL; /*указател към масива от указатели */
    char buf[255];
    FILE *fp=NULL;
    int i,size=DELTA,num=0;
    int ok; /* за евентуално подреждане в азбучен ред на низовете */
    if((ptr = (char **)malloc(DELTA*sizeof(char*)))==NULL) { /* създава се масив от DELTA указателя */
        printf("Memory allocation error!"); return 1 ;
    }
    if((fp=fopen("test.txt", "r")) == NULL) { /* файлът от който се чете */
        printf("Cannot open file.\n"); return 2;
    }
    for(;fgets(buf,255,fp);) { /* чете се следващият ред в низ */
        ptr=include(buf,ptr,&size,&num); /* низът се закачва */
        if(ptr==NULL) { /* ако е неуспешно */
            printf("Memory allocation error!");
            fclose(fp); return 1;
        }
    }
    /* извеждане на екрана прочетените редове, броя на редовете и размера на масива от указатели
    for(i=0;i<num;i++){
        puts(ptr[i]);
    }
    printf("\n found %d lines; in the array there are %d pointers\n",num,size);

/* обработка на масива - например подреждане на низовете в азбучен ред */
do{
    ok = 1;
    for(i=0;i<num-1;i++){
        if(strcmp(ptr[i],ptr[i+1])>0){
            /*смяна местата на указателите */
            char * tmp = ptr[i];
            ptr[i]=ptr[i+1];
            ptr[i+1]=tmp;
            ok=0;
        }
    }
}while (!ok);
/* извеждане на екрана подредените по азбучен ред низове */
for(i=0;i<num;i++){
    puts(ptr[i]);
}

/* освобождаване на динамично резервираната памет */
for(i=0;i<num;i++){
    free(ptr[i]); /* трябва да се освободи паметта за отделните редове */
}
free(ptr); /* освобождаване на масива от указатели */
fclose(fp);
return 0;
}

```