

Пример с динамичен масив от низове всеки ред от файла се записва в отделен низ

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define DELTA 5
char** include(char *bf,char** p,int *sizeAd,int* nm)
{
    char **tmp;
    if(*nm<*sizeAd) {
        if ((p[*nm]= (char*)malloc(strlen(bf)+1))==NULL) return NULL; /* Запазва се място за реда и се копира */
        strcpy(p[(*nm)++],bf);
    }
    else {
        *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) {
            p=tmp;
            if((p[*nm]= (char*)malloc(strlen(bf)+1))==NULL) return NULL; /* Запазва се място за реда и се копира */
            strcpy(p[(*nm)++],bf);
        }
        else {
            free(p);
            p=NULL;
        }
    }
    return p;
}
int main(){
    char **ptr;
    char buf[255];
    FILE *fp=NULL;
    int size=DELTA,num=0;
    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) {
        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) {
            printf("Memory allocation error!");
            fclose(fp); return 1;
        }
    }
    /* Обработка на получения масив */
    for(int i=0;i<num;i++){
        puts(ptr[i]);
        free(ptr[i]); /* трябва да се освободи паметта за отделните редове */
    }
    printf("\n found %d lines total size is %d \n",num,size);
    free(ptr); fclose(fp);
    return 0;
}
```