

Tölvunarfræði 2 - Heimadæmi 01

Teitur Guðmundarson

1. Fyrir tvívíð fylki er oft einfaldara að geyma þau í einu löngu fylki, þar sem hver lína í fylkinu kemur á eftir annarri. Við skilgreinum eftirfarandi struct til að halda utan $n \times m$ fylki A , breytan a verður þá að vera einvítt fylki af lengd nm , stakið $A_{i,j}$ er þá í sæti $(i-1)m + j - 1$ (hér byrja fylki í stærðfræði í línu 1 en í tölvunarfræði byrjar allt á staðsetningunni 0).

```
typedef struct matrix {  
double* a;  
int n;  
int m;  
} matrix;
```

Skrifið fall sem tekur tvö fylki A og B og reiknar út fylkjamargföldunina AB , fallið ykkar á að taka frá minni fyrir útkomuna en þarf ekki að skila því til baka. Skilið einungis fallinu ykkar.

```
matrix mult(matrix A, matrix B) {  
    matrix C;  
    C.a = malloc(sizeof(double)*A.n*B.m);  
    C.n = A.n;  
    C.m = B.m;  
    // triple for loop for radir A.n dalkar B.m stak C.n  
    for (int i = 0; i < A.n; i++) {  
        for (int j = 0; j < B.m; j++) {  
            for (int k = 0; k < A.m; k++) {  
                C.a[j*C.m+i] += A.a[i*A.m+k]*B.a[k*C.m+j];  
            } printf("%.2f ", C.a[j*C.m+i]);  
        } printf("\n");  
    }  
    return C;  
}
```

```

2. #include <stdio.h>
   #include <stdlib.h>

   typedef struct matrix {
       double* a;
       int n;
       int m;
   } matrix;

   matrix mult(matrix A, matrix B) {
       matrix C;
       C.a = malloc(sizeof(double)*A.n*B.m);
       C.n = A.n;
       C.m = B.m;
       for (int i = 0; i < A.n; i++) {
           for (int j = 0; j < B.m; j++) {
               for (int k = 0; k < A.m; k++) {
                   C.a[j*C.m+i] += A.a[i*A.m+k]*B.a[j*B.n+k];
               } printf("%.1f ", C.a[j*C.m+i]);
           } printf("\n");
       }
       return C;
   }

   int main() {
       matrix A;
       matrix B;
       matrix C;
       int n, m;
       float a[20];
       scanf("%d %d", &n, &m);
       for (int i = 0; i < (n*m); i++) {
           scanf("%f", &a[i]);
       }
       A.n = n;
       A.m = m;
       A.a = malloc((n*m)*sizeof(double));
       for (int i = 0; i < (n*m); i++) {
           A.a[i] = a[i];
       }

       scanf("%d %d", &n, &m);
       for (int i = 0; i < (n*m); i++) {
           scanf("%f", &a[i]);
       }

       B.n = n;
       B.m = m;
       B.a = malloc((n*m)*sizeof(double));
       for (int i = 0; i < (n*m); i++) {
           B.a[i] = a[i];
       }

       mult(A, B);
   }

```

3. Forritið `mycat.c` tekur inn eina skrá sem viðfang og skrifar út á staðalúttak. Breytið forritinu þannig að það skrifi allar skrár sem koma inn sem viðfang og skrifa út á staðalúttak í þeirri röð sem þau komu fyrir. Ef engin skrá er gefin, eða skráarnafnið - er gefið sem eina viðfang þá á að lesa af staðalinntaki og skrifa á staðalúttak. *Vísbending:* það einfaldar kóðan að líta á staðalinntak sem skrá, sjá umfjöllun um `stdin`

```
mycat.c
#include <stdio.h>

int main(int argc, char** argv) {
    int counter = 1;
    while(counter < argc) {
        // fyrsta viðfang
        char* filename = argv[1];
        // opnar skrá til lesturs
        FILE* file = fopen(filename, "r");
        char line[1024]; // line er char fylki sem við lesum inn í

        // fgets les eina línu í einu
        while (fgets(line, sizeof(line), file)) {
            printf("%s", line);
        }
        // lokum skránni
        fclose(file);
        counter++;
    }
}
```

4. Skrifðið forritið `mygrep.c` sem tekur inn eitt viðfang sem er strengur `pattern` (án bila), 0 eða fleiri skrár og prentar einungis þær línur sem innihalda strenginn `pattern` sem hlutstreng. Hér má ekki nota nein forritasöfn til að framkvæma strengjaleitina, þið verðið að kóða það sjálf. Skilið forritinu ykkar, keyrið það á skránni `holmes.txt` og teljið fjölda lína þar sem strengurinn `Holmes` kemur fyrir (flettið upp á forritinu `wc`).

```
mygrep.c
#include <stdio.h>
#include <string.h>

int my_grep(char* line, char* pattern) {
    for (int i = 0, k = 0; i < strlen(line); i++) {
        if (k == strlen(pattern))
            return 1;
        if (pattern[k] == line[i])
            k++;
        else k = 0;
    }
    return 0;
}

int main(int argc, char** argv) {
    int counter = 2;
    char* pattern = argv[1];
    while (counter < argc) {
        // fyrsta viðfang
        char* filename = argv[counter];
        // opnar skrá til lesturs
        FILE* file = fopen(filename, "r");
        char line[1024];

        // fgets les eina línu í einu
        while (fgets(line, sizeof(line), file)) {
            if (my_grep(line, pattern) == 1)
                printf("%s", line);
        }
        // lokum skránni
        fclose(file);
        counter++;
    }
    return 0;
}
```

5. Bætið við eftirfarandi föllum við `vector.c` skrána. Öll föllin taka bendi á `vector` sem fyrsta viðfang.

- `vector_set` - tekur sæti `i` og gildi `x`, virkar eins og `v[i] = x`
- `vector_remove` - tekur sæti `i` úr `vectornum` og skilar gildinu sem var í því sæti. Athugið að `vectorinn` minnkar í stærð um einn og gildin í fylkinu gætu hliðrast til
- `vector_copy` - afritar `vectorinn` í nýjan `vector`
- `vector_find` - tekur inn stak `x` og skilar fyrsta sætinu þar sem `x` kemur fyrir eða -1 ef það kemur ekki fyrir í `vectornum`.

`vector.c`
