

Řešení sady 1

Úvod do programování 2
Tomáš Kühr

Maximum dvojrozměrného pole

```
#define SLOUPCU 4

int maximum(int prvky[][SLOUPCU], int radku) {
    int i, j;
    int max = INT_MIN;

    for (i = 0; i < radku; i++) {
        for (j = 0; j < SLOUPCU; j++) {
            if (prvky[i][j] > max) {
                max = prvky[i][j];
            }
        }
    }
    return max;
}
```

Suma řádků dvojrozměrného pole

```
#define SLOUPCU 4
int *suma_radku(int prvky[][SLOUPCU], int radku) {
    int i, j;
    int *vysledek;

    vysledek = (int *) malloc(radku * sizeof(int));
    if (vysledek == NULL) return NULL;

    for (i = 0; i < radku; i++) {
        vysledek[i] = 0;
        for (j = 0; j < SLOUPCU; j++) {
            vysledek[i] += prvky[i][j];
        }
    }
    return vysledek;
}
```

Četnost znaku v poli řetězců

```
int vyskyty(char *texty[], int pocet, char hledany) {  
    int vysledek = 0;  
    char *radek;  
  
    for (int i = 0; i < pocet; i++) {  
        radek = texty[i];  
        while(*radek) {  
            if (*radek == hledany) {  
                vysledek++;  
            }  
            radek++;  
        }  
    }  
    return vysledek;  
}
```

Součet matic

```
double **soucet(int m, int n, double **A, double **B) {
    double **vysledek;
    int i, j, k;
    vysledek = (double **) malloc(m * sizeof(double *));
    if (vysledek == NULL) return NULL;
    for (i = 0; i < m; i++){
        vysledek[i] = (double *) malloc(n * sizeof(double));
        if (vysledek[i] == NULL){ // uklid
            for (k = 0; k < i; k++) free(vysledek[k]);
            free(vysledek);
            return NULL;
        }
        for (j = 0; j < n; j++){
            vysledek[i][j] = A[i][j] + B[i][j];
        }
    }
    return vysledek;
}
```

Součin matic

```
double **soucin(int m, int n, int o, double **A, double **B) {
    double **vysledek;
    int i, j, k, l;
    vysledek = (double **) malloc(m * sizeof(double *));
    if (vysledek == NULL) return NULL;
    for (i = 0; i < m; i++){
        vysledek[i] = (double *) malloc(o * sizeof(double));
        if (vysledek[i] == NULL){ // uklid
            for (l = 0; l < i; l++) free(vysledek[l]);
            free(vysledek); return NULL;
        }
        for (j = 0; j < o; j++){
            vysledek[i][j] = 0;
            for (k = 0; k < n; k++){
                vysledek[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    return vysledek;
}
```