

Řešení sady 4

Úvod do programování 1

Studenti - varianta 1

```
typedef struct{
    char den; char mesic; short rok;
} datum;
```

```
typedef struct{
    char jmeno[20]; char prijmeni[20]; datum narozen;
} student;
```

```
int porovnej_vek(student s1, student s2){
    if (s1.narozen.rok < s2.narozen.rok) return -1;
    if (s1.narozen.rok > s2.narozen.rok) return 1;
    if (s1.narozen.mesic < s2.narozen.mesic) return -1;
    if (s1.narozen.mesic > s2.narozen.mesic) return 1;
    if (s1.narozen.den < s2.narozen.den) return -1;
    if (s1.narozen.den > s2.narozen.den) return 1;
    return 0;
}
```

Studenti - varianta 2

```
int porovnej_vek2(student s1, student s2){  
    int prvni=s1.narozen.rok*10000+s1.narozen.mesic*100+s1.narozen.den;  
    int druhy=s2.narozen.rok*10000+s2.narozen.mesic*100+s2.narozen.den;  
    int vysledek = (prvni > druhy) ? 1 : ((prvni < druhy) ? -1 : 0);  
    return vysledek;  
}
```

```
int main(){  
    student a = { "Adam", "Novak", { 1, 1, 2014 } };  
    student b = { "Petr", "Stary", { 31, 11, 2013 } };  
  
    int vysl1 = porovnej_vek(a, b);  
    int vysl2 = porovnej_vek2(a, b);  
    printf("%i %i\n", vysl1, vysl2);  
    return 0;  
}
```

Součet zlomků

```
typedef struct{
    int citatel; int jmenovatel;
} zlomek;

int nsd(int a, int b){
    while (b){ int r = a % b; a = b; b = r; }
    return a;
}

zlomek soucet(zlomek a, zlomek b){
    zlomek vysl;
    int kraceni;
    vysl.citatel = a.citatel * b.jmenovatel + b.citatel * a.jmenovatel;
    vysl.jmenovatel = a.jmenovatel * b.jmenovatel;

    kraceni = nsd(abs(vysl.citatel), abs(vysl.jmenovatel));
    vysl.citatel /= kraceni;
    vysl.jmenovatel /= kraceni;

    if (vysl.jmenovatel < 0){vysl.citatel *= -1; vysl.jmenovatel *= -1; }
    return vysl;
}
```

Porovnání textových řetězců

```
int porovnej(char *t1, char *t2){
    while (*t1 || *t2){ // pokud nejsou oba na konci
        if (*t1 < *t2) return -1;
        if (*t1 > *t2) return 1;
        t1++;
        t2++;
    }
    return 0;
}
```

```
int main(){
    char str1[] = "adc";
    char str2[] = "abcd";

    printf("%i\n", porovnej(str1, str2));

    return 0;
}
```

Hledání podřetězce zprava

```
char *strrstr(const char *text, const char *hledany){
    char *temp;
    temp = text + strlen(text) - strlen(hledany); // odtud zacneme

    while (temp >= text){ // temp je porad nekde v rezezci text
        char *p_temp = temp; // pomocny ukazatel pro porovnani
        char *p_hledany = hledany; // pomocny ukazatel pro porovnani
        while (*p_temp == *p_hledany){ // dokud jsou znaky stejne
            p_temp++; // jdeme porovnat dalsi znak
            p_hledany++; // i v druhem retezci
            if (!*p_hledany) // jsme na konci hledaneho retezce
                return temp;
        }
        temp--; // tady jsme nenasli, jdeme o 1 znak doleva
    }
    return NULL;
}
```