

Avditorne vaje za Programiranje II

Matej Blagšič

20. marec 2018

Kazalo

1	Prva vaja	2
2	Druga vaja	2
3	Tretja vaja	3
4	Četrta vaja	4

1 Prva vaja

2 Druga vaja

Izračunaj $\int_{x_0}^{x_1} 2x^2 - 5x \, dx$. Rezultat preveri analitično.

Če analitično integriramo itegral, dobimo: $\int_{x_0}^{x_1} 2x^2 - 5x \, dx = 2\frac{x_1^3}{3} - 5\frac{x_0^2}{2}$

Sedaj spišimo kodo:

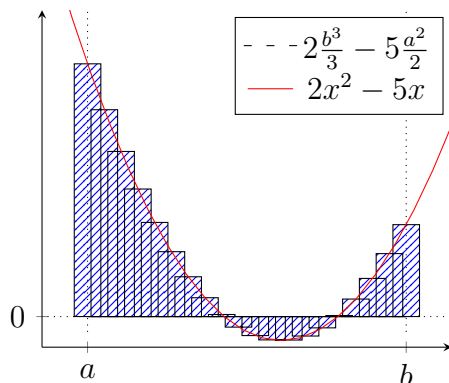
```
int main(){
    float x, x0, x1;
    float dx = 0.0000001;
    float integral = 0;
    printf("vnesi spodno mejo");
    scanf("%f",&x0);
    printf("Vnesi zgornjo mejo");
    scanf("%f",&x1);

    for(x=x0; x<x1;x+=dx){
        integral += dx*(2*x*x-5*x);
    }

    printf("Integral znasa: %f\n", integral);
    return 0;
}
```

Pri programu nam spremenljivka dx sporoči, kako širok del območja integrira. Manjša, kot je cifra, bolj natančno izračuna. x0 in x1 sta spodnja in zgornja meja integracije, x pa je spremenljivka, ki jo premikamo po intervalu za dx razdaljo in seštevamo pravokotnike.

Integracija funkcije na območju od a do b



3 Tretja vaja

1. naloga

Napiši program, ki izpiše in izračuna faktorielo(fakulteta) nekega števila(1-20)

Pri temu programu spoznamo omejitve velikosti spremenljivk. Pomembno je, da so števila, ki jih hočemo hraniti in predstavljani v polni natančnosti ne presegajo velikosti spremenljivke. Če ne, potem se prične pačenje podatkov. Vidimo, da lahko uporabimo izrad long long in s tem povečamo obseg navadnega long tipa. Podobno lahko naredimo tudi s spremenljivkam s plavajočo vejico, recimo `long double`.

```
int main(){
    unsigned long long resitev = 1, n;

    for(n=1; n<=20; n++){
        for(int i=1; i<=n; i++){
            resitev *= i;
        }
        printf("Faktoriela od %lld! je %lld\n", n, resitev);
        resitev = 1;
    }
    return 0;
}
```

2. naloga

Napiši program, ki sešteje maso lune in zemlje ter sonca.

Namen vaje je dodatno spoznati omejitve spremenljivk v programskem jeziku C. Mase lune, zemlje in sonca so ogromne, zato se začenja poznati popačenje podatkov. Ugotovili smo, da dobimo kar se da dobre rezultate, če uporabimo `double`, ki ima največji obseg, a se vseeno na določenem mestu pojavi naključna številka, ki ni podana med vhodnimi podatki.

```
int main(){
    double luna = 7.348e22;
    double zemlja = 5.972e24;
    double sonce = 1.989e30;

    printf("Masa lune je: %40.0lf \n",luna);
    printf("Masa zemlje je: %40.0lf \n",zemlja);
    printf("Masa sonca je: %40.0lf \n",sonce);
    printf("Skupna masa je: %40.0lf\n", zemlja+luna);
    printf("Skupna masa sonca pa lune je: %40.0lf\n", sonce+luna);

    return 0;
}
```

4 Četrta vaja

1. naloga

Napiši funkcijo, ki sešteje dve razdalji podani v čevljih in palcih(feet and inches)

Vemo, da je 1 čevlj 12 palcev. To potrebujemo, da pretvorimo palce v čevlje, kajti če pri vsoti dobimo recimo 13 palcev, pretvorimo to v en čevlj in en palec.

```

#include <stdio.h>
struct razdalja{
    int foot;
    int inch;
};
struct razdalja seštevanje(struct razdalja x, struct razdalja y);
int main(){
    struct razdalja a, b, r;
    printf("Vnesi prvo razdaljo v obliki a b ");
    scanf("\n %d\'%d\'", &a.foot, &a.inch);
    printf("Vnesi drugo razdaljo v obliki a b ");
    scanf("\n %d\'%d\'", &b.foot, &b.inch);
    r = seštevanje(a, b);
    printf("%d %d\n", r.foot, r.inch);
    return 0;
}
struct razdalja seštevanje(struct razdalja x, struct razdalja y){
    struct razdalja z;
    z.inch = x.inch + y.inch;
    z.foot = x.foot + y.foot;
    if(z.inch >=12){
        z.foot++;
        z.inch -=12;
    }
    return z;
}

```

Vidimo, da smo definirali novo funkcijo za seštevanje, katere tip je enak izhodnemu podatku, torej novemu tipu razdalja, ki jo definiramo s struct razdalja.

2. naloga

Imamo program, ki nas nauči o vrstah konstant pri primerjavah.

V temu programu spoznamo, da v C-ju so vse konstante tipa `double`. To je zelo pomembno. Poglejmo si priložen program. Hočemo primerjati `x` z njegovo vrednostjo. Program nam vrne nič, kot da nista iste, čeprav sta. Naš dvojni enačaj je tipa `int`, a konstanta `0.2`, ki se primerja z `x`, je pa tipa `double` in ne `float`, kot smo hoteli definirati `x`. Zato popravimo spremenljivko `x` v `double`.

```
#include <stdio.h>

int main(){
    double x = 0.2; //prej: float x = 0.2;
    printf("%d\n", x == 0.2);
    return 0;
}
```