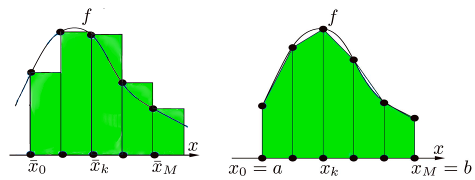



Rysunek 1: Całkowanie metodą prostokątów-lewostronną oraz całkowanie metodą trapezów

# 6. Obliczanie całek (wskaźniki do funkcji)

## Program 6.1: Całki jednowymiarowe - metoda prostokątów lewostronna

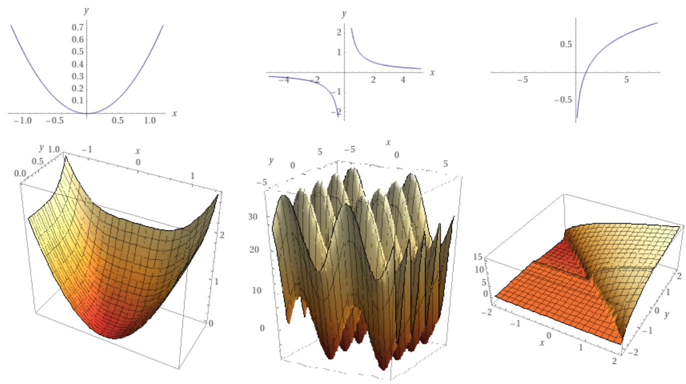
Szablon programu należy uzupełnić o definicję funkcji `integr_rectleft(...)`, która obliczy całkę z funkcji $x^2/2$. Ma to być całka wykorzystująca metodę prostokątów-lewostronna. Przeczytaj o wskaźnikach do funkcji na stronie 63 wykładu: `https://home.agh.edu.pl/~pszwed/wiki/lib/exe/fetch.php?media=06-imperatywne-jezyk-c-wskazniki.pdf`

- **Wejście**
  1 `x1(dolna granica całkowania) x2(granica górna) n(liczba kroków całkowania między przedziałami)`
- **Wyjście**
  Wartość całki oznaczonej z funkcji $x^2/2$
- **Przykład:**
  Wejście: 1 0 10 1000
  Wyjście: 166.42

## Program 6.2: Całki jednowymiarowe - metoda prostokątów i trapezów

Szablon programu należy uzupełnić o definicję funkcji `int_rightrect(...)`, która oblicza całkę funkcji metodą prostokątów-prawostronną, `int_midrect(...)`, obliczającą całkę metodą prostokątów względem środka prostokąta oraz `int_trapez(...)`, która oblicza całkę metodą trapezów. Funkcje podcałkowe to cztery funkcje: $f(x) = x$, $f(x) = x^2/2$, $f(x) = 1/x$ oraz $f(x) = log_{10}(x)$ (Rys.2 górny). Należy również uzupełnić definicję typów danych dla funkcji całkujących jak i dla funkcji podcałkowych. Wykresy po wpisaniu funkcji można obejrzeć na stronie: `https://www.wolframalpha.com/`

- **Wejście**
  2 `x1(dolna granica) x2(górna granica) n(liczba kroków)`



Rysunek 2: Funkcje podcałkowe

- **Wyjście**
  Wartość trzech ww. rodzajów całek oznaczonych dla czterech ww. funkcji podcałkowych.

- **Przykład:**
  Wejście: 2 0 10 1000
  Wyjście: 50.05 50.00 50.00 166.92 166.67 166.67 7.49 8.87 inf 5.68 5.66 -inf

## Program 6.3: Całki dwuwymiarowe

Szablon programu należy uzupełnić o definicję funkcji `double_integr(...)`, która oblicza podwójna całkę funkcji metodą prostokątów-lewostronną. Funkcje podcałkowe to trzy funkcje: $f(x,y) = \sqrt{x^4 + y^5}$, $f(x,y) = 8sin(x) + (5cos(y))^2$, $f(x,y) = 9\sqrt{x+y} - 2\sqrt{y}$ (Rys.2 dolny). Uzupełnij definicję typów danych dla funkcji podcałkowych.

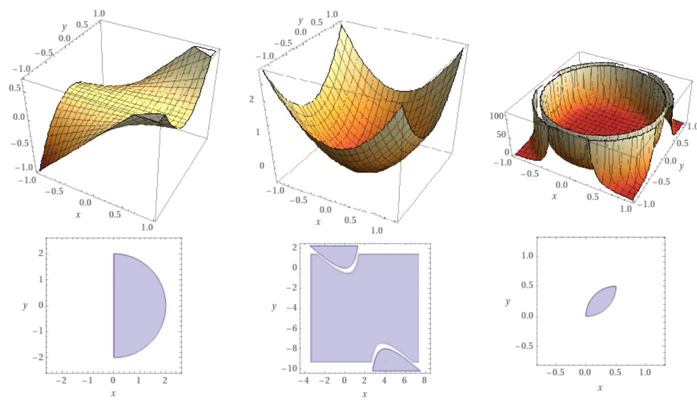
- **Wejście**
  `3 x1 x2 nx y1 y2 ny`

- **Wyjście**
  Wartość całki oznaczonej dla trzech ww. funkcji podcałkowych.

- **Przykład:**
  Wejście: 3 1 2 1000 2 3 1000
  Wyjście: 10.45 23.12 14.82

## Zadanie 6.4. Całki trójwymiarowe

Szablon programu należy uzupełnić o definicję funkcji `integr3D(...)`, która oblicza potrójną całkę funkcji metodą prostokątów-lewostronną. Funkcje podcałkowe to dwie funkcje: $f(x,y,z) = x + 3y + 5z$, $f(x,y,z) = 8sin(x) + (5cos(y))^2 + 2z$. Uzupełnij definicję typów danych dla funkcji podcałkowych.



Rysunek 3: Funkcje podcałkowe oraz ich ograniczenia poniżej funkcji

- **Wejście**
  `3 x1 x2 nx y1 y2 ny z1 z2 nz`
- **Wyjście**
  Wartość całki oznaczonej dla dwóch ww. funkcji podcałkowych.
- **Przykład:**
  Wejście:
  4 1 2 200 2 3 200 3 4 200
  Wyjście:
  26.48 30.08

## Program 6.5: Całki dwuwymiarowe ze wskaźnikiem do predykatu boundary

Szablon programu należy uzupełnić o definicję funkcji `doub_int_bound(...)`, która jest całką podwójną opartą na metodzie prostokątów, lewostronną uwzględniającą ograniczenia wynikające z funkcji "boundary". Funkcje podcałkowe to trzy funkcje: (1) $xy^2$, (2) $x^2 + y^2$, (3) $1/(1 - x^2 - y^2)^2$. Odpowiadające icm trzy funkcje ograniczeńc "boundary" to: (1) $x^2+y^2 <= 4, x >= 0$, (2) $1 <= x^2+xy-2y <= 0$, (3) $x^2 + y^2 <= x, x^2 + y^2 <= y$ (Rys.3). Uzupełnij definicję typów danych dla funkcji podcałkowych.

- **Wejście**
  `5 x1 x2 nx y1 y2 ny`
- **Wyjście**
  Wartość całki oznaczonej dla trzech ww. funkcji podcałkowych.
- **Przykład:**
  Wejście:

5 -2 2 1000 -2 2 1000
Wyjście:
4.27 40.20 0.21

### Program 6.6: Zadanie dodatkowe (*)

Zrealizuj całkę trójwymiarową uwzględniającą ograniczenia "boundary" zgodnie z pseudokodem zamieszczonym poniżej:

---

```
def integrate3D(f,boundary,rangex=[0,1],rangey=[0,1],rangez=[0,1],n=1000):
    sum=0
        for x in np.linspace(rangex[0],rangex[1],n):
        for y in np.linspace(rangey[0],rangey[1],n):
            for z in np.linspace(rangez[0],rangez[1],n):
                if boundary(x,y,z):
                    sum+=f(x,y,z)
    return sum/n**3
```