

```

{
  "cells": [
    {
      "attachments": {},
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "1. Faça um programa que determine e mostre os cinco primeiros  
múltiplos de 3,\n",
        "considerando números maiores que 0."
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 1,
      "metadata": {},
      "outputs": [
        {
          "name": "stdout",
          "output_type": "stream",
          "text": [
            "3\n",
            "6\n",
            "9\n",
            "12\n",
            "15\n"
          ]
        }
      ],
      "source": [
        "for n in range(1,6):\n",
        "    print(n*3)"
      ]
    },
    {
      "attachments": {},
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "2. Escreva um programa que escreva na tela, de 1 até 100, de 1  
em 1, 2 vezes. A primeira vez deve usar a estrutura de repetição for  
e a segunda while."
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 7,
      "metadata": {},
      "outputs": [
        {
          "name": "stdout",
          "output_type": "stream",
          "text": [
            "1\n",

```

"2\n",
"3\n",
"4\n",
"5\n",
"6\n",
"7\n",
"8\n",
"9\n",
"10\n",
"11\n",
"12\n",
"13\n",
"14\n",
"15\n",
"16\n",
"17\n",
"18\n",
"19\n",
"20\n",
"21\n",
"22\n",
"23\n",
"24\n",
"25\n",
"26\n",
"27\n",
"28\n",
"29\n",
"30\n",
"31\n",
"32\n",
"33\n",
"34\n",
"35\n",
"36\n",
"37\n",
"38\n",
"39\n",
"40\n",
"41\n",
"42\n",
"43\n",
"44\n",
"45\n",
"46\n",
"47\n",
"48\n",
"49\n",
"50\n",
"51\n",
"52\n",
"53\n",
"54\n",
"55\n",

```
"56\n",
"57\n",
"58\n",
"59\n",
"60\n",
"61\n",
"62\n",
"63\n",
"64\n",
"65\n",
"66\n",
"67\n",
"68\n",
"69\n",
"70\n",
"71\n",
"72\n",
"73\n",
"74\n",
"75\n",
"76\n",
"77\n",
"78\n",
"79\n",
"80\n",
"81\n",
"82\n",
"83\n",
"84\n",
"85\n",
"86\n",
"87\n",
"88\n",
"89\n",
"90\n",
"91\n",
"92\n",
"93\n",
"94\n",
"95\n",
"96\n",
"97\n",
"98\n",
"99\n",
"100\n"
]
}
],
"source": [
    "# Usando a estrutura de repetição for\n",
    "for n in range(1,101):\n",
    "    print(n)"
]
},
```

```
{
  "cell_type": "code",
  "execution_count": 9,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "1\n",
        "2\n",
        "3\n",
        "4\n",
        "5\n",
        "6\n",
        "7\n",
        "8\n",
        "9\n",
        "10\n",
        "11\n",
        "12\n",
        "13\n",
        "14\n",
        "15\n",
        "16\n",
        "17\n",
        "18\n",
        "19\n",
        "20\n",
        "21\n",
        "22\n",
        "23\n",
        "24\n",
        "25\n",
        "26\n",
        "27\n",
        "28\n",
        "29\n",
        "30\n",
        "31\n",
        "32\n",
        "33\n",
        "34\n",
        "35\n",
        "36\n",
        "37\n",
        "38\n",
        "39\n",
        "40\n",
        "41\n",
        "42\n",
        "43\n",
        "44\n",
        "45\n",

```

"46\n",
"47\n",
"48\n",
"49\n",
"50\n",
"51\n",
"52\n",
"53\n",
"54\n",
"55\n",
"56\n",
"57\n",
"58\n",
"59\n",
"60\n",
"61\n",
"62\n",
"63\n",
"64\n",
"65\n",
"66\n",
"67\n",
"68\n",
"69\n",
"70\n",
"71\n",
"72\n",
"73\n",
"74\n",
"75\n",
"76\n",
"77\n",
"78\n",
"79\n",
"80\n",
"81\n",
"82\n",
"83\n",
"84\n",
"85\n",
"86\n",
"87\n",
"88\n",
"89\n",
"90\n",
"91\n",
"92\n",
"93\n",
"94\n",
"95\n",
"96\n",
"97\n",
"98\n",
"99\n",

```

        "100\n"
    ]
}
],
"source": [
    "# Usando a estrutura de repetição while\n",
    "\n",
    "x=1\n",
    "while(x <= 100):\n",
    "    print(x)\n",
    "    x+=1"
]
},
{
    "attachments": {},
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "3. Faça um algoritmo utilizando o comando while que mostra uma contagem regressiva na tela, iniciando em 10 e terminando em 0. Mostrar uma mensagem "FIM!" após a contagem."
    ]
},
{
    "cell_type": "code",
    "execution_count": 4,
    "metadata": {},
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "CONTAGEM REGRESSIVA:\n",
                "10\n",
                "9\n",
                "8\n",
                "7\n",
                "6\n",
                "5\n",
                "4\n",
                "3\n",
                "2\n",
                "1\n",
                "0\n",
                "FIM!\n"
            ]
        }
    ]
},
"source": [
    "print(\"CONTAGEM REGRESSIVA:\")\n",
    "n = 10\n",
    "while (n >= 0):\n",
    "    print(n)\n",
    "    n-=1\n",

```

```

    "print(\"FIM!\")"
  ],
  {
    "attachments": {},
    "cell_type": "markdown",
    "metadata": {},
    "source": [
      "4. Escreva um programa que declare um inteiro, inicialize-o com 0, e incremente-o de 1000 em 1000, imprimindo seu valor na tela, até que seu valor seja 100000 (cem mil)."]
    ],
    {
      "cell_type": "code",
      "execution_count": 5,
      "metadata": {},
      "outputs": [
        {
          "name": "stdout",
          "output_type": "stream",
          "text": [
            "0\n",
            "1000\n",
            "2000\n",
            "3000\n",
            "4000\n",
            "5000\n",
            "6000\n",
            "7000\n",
            "8000\n",
            "9000\n",
            "10000\n",
            "11000\n",
            "12000\n",
            "13000\n",
            "14000\n",
            "15000\n",
            "16000\n",
            "17000\n",
            "18000\n",
            "19000\n",
            "20000\n",
            "21000\n",
            "22000\n",
            "23000\n",
            "24000\n",
            "25000\n",
            "26000\n",
            "27000\n",
            "28000\n",
            "29000\n",
            "30000\n",
            "31000\n",

```

"32000\n",
"33000\n",
"34000\n",
"35000\n",
"36000\n",
"37000\n",
"38000\n",
"39000\n",
"40000\n",
"41000\n",
"42000\n",
"43000\n",
"44000\n",
"45000\n",
"46000\n",
"47000\n",
"48000\n",
"49000\n",
"50000\n",
"51000\n",
"52000\n",
"53000\n",
"54000\n",
"55000\n",
"56000\n",
"57000\n",
"58000\n",
"59000\n",
"60000\n",
"61000\n",
"62000\n",
"63000\n",
"64000\n",
"65000\n",
"66000\n",
"67000\n",
"68000\n",
"69000\n",
"70000\n",
"71000\n",
"72000\n",
"73000\n",
"74000\n",
"75000\n",
"76000\n",
"77000\n",
"78000\n",
"79000\n",
"80000\n",
"81000\n",
"82000\n",
"83000\n",
"84000\n",
"85000\n",


```

        "86000\n",
        "87000\n",
        "88000\n",
        "89000\n",
        "90000\n",
        "91000\n",
        "92000\n",
        "93000\n",
        "94000\n",
        "95000\n",
        "96000\n",
        "97000\n",
        "98000\n",
        "99000\n",
        "100000\n",
        "CEM MIL!\n"
    ]
}
],
"source": [
    "n = 0\n",
    "while (n <= 100000):\n",
    "    print(n)\n",
    "    n+=1000\n",
    "print(\"CEM MIL!\")"
]
},
{
    "attachments": {},
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "5. Faça um programa que peça ao usuário para digitar 10 valores  
e some-os."
    ]
},
{
    "cell_type": "code",
    "execution_count": 19,
    "metadata": {},
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "A soma dos 10 valores digitados eh: 25.0\n"
            ]
        }
    ]
},
"source": [
    "x=1\n",
    "y=0\n",
    "while (x<=10):\n",
    "    y = y + float(input(\"Digite um número:\"))\n",

```

```

        "    x+=1\n",
        "\n",
        "print(\"A soma dos 10 valores digitados eh:\", y)"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "6. Faça um programa que leia 10 inteiros e imprima sua média."
    ]
},
{
    "cell_type": "code",
    "execution_count": 24,
    "metadata": {},
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "A média dos 10 numeros inteiros digitados eh: -1.0\n"
            ]
        }
    ],
    "source": [
        "x=1\n",
        "y=0\n",
        "while (x<=10):\n",
        "    y = y + int (input(\"Digite um número inteiro:\"))\n",
        "    x+=1\n",
        "\n",
        "print(\"A média dos 10 números inteiros digitados é:\", y/10)"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
},
{
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "7. Faça um programa que leia 10 inteiros positivos, ignorando não positivos, e imprima sua média."
    ]
},
{
    "cell_type": "code",
    "execution_count": 32,
    "metadata": {},

```

```

"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "A média dos 10 numeros inteiros digitados é: 1.0\n"
    ]
  }
],
"source": [
  "x=1\n",
  "y=0\n",
  "soma=0\n",
  "for i in range(0,10):\n",
  "    while (y<=0):\n",
  "        y = int (input(\"Digite um número inteiro positivo:\"))\n",
  "\n",
  "        if y>0:\n",
  "            soma = soma + y\n",
  "            continue\n",
  "        y = 0\n",
  "\n",
  "print(\"A média dos 10 números inteiros positivos digitados é:\n", soma/10)"
]
},
{
  "cell_type": "markdown",
  "metadata": {},
  "source": [
    "8. Escreva um programa que leia 10 números e escreva o menor valor lido e o maior valor lido."
  ]
},
{
  "cell_type": "code",
  "execution_count": 38,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "0 maior valor digitado foi 3.0 e o menor valor digitado foi 1.0\n"
      ]
    }
  ]
},
"source": [
  "x = float(input(\"Digite 10 números:\\n\\n\"))\n",
  "menor = x\n",
  "maior = x\n",
  "for i in range(1,10):\n",
  "    x = float(input())\n",

```

```

        "    if x < menor:\n",
        "        menor = x\n",
        "    if x > maior:\n",
        "        maior = x\n",
        "print(\"0 maior valor digitado foi:\",maior,\"e o menor valor\n",
        "digitado foi:\",menor)\"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "9. Faça um programa que leia um número inteiro N e depois\n",
        "imprima os N primeiros números naturais ímpares."
    ]
},
{
    "cell_type": "code",
    "execution_count": 46,
    "metadata": {},
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "0s 9 primeiros números naturais ímpares são:\n",
                "1\n",
                "3\n",
                "5\n",
                "7\n",
                "9\n",
                "11\n",
                "13\n",
                "15\n",
                "17\n"
            ]
        }
    ],
    "source": [
        "n = int(input(\"Digite um número inteiro:\"))\n",
        "print(\"0s\",n, \"primeiros números naturais ímpares são:\")\n",
        "\n",
        "for i in range(1,n+1):\n",
        "    print(i*2-1)"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "10. Faça um programa que calcule e mostre a soma dos 50\n",
        "primeiros números pares."
    ]
},

```

```

{
  "cell_type": "code",
  "execution_count": 56,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "A soma dos 50 primeiros números pares é: 2550\n"
      ]
    }
  ],
  "source": [
    "soma = 0\n",
    "for pares in range(1,51):\n",
    "    soma = soma + pares*2\n",
    "print(\"A soma dos 50 primeiros números pares é:\",soma)"
  ]
},
{
  "metadata": {
    "kernel_spec": {
      "display_name": "base",
      "language": "python",
      "name": "python3"
    },
    "language_info": {
      "codemirror_mode": {
        "name": "ipython",
        "version": 3
      },
      "file_extension": ".py",
      "mimetype": "text/x-python",
      "name": "python",
      "nbconvert_exporter": "python",
      "pygments_lexer": "ipython3",
      "version": "3.8.5"
    },
    "orig_nbformat": 4,
    "vscode": {
      "interpreter": {
        "hash":
"f49206fcf84a9145e7e21228cbafa911d1ac18292303b01e865d8267a9c448f7"
      }
    }
  },
  "nbformat": 4,
  "nbformat_minor": 2
}

```