|  |  |  |
| --- | --- | --- |
| Departamento de Matemática da Universidade de Coimbra | | |
| **2021/2022** | Métodos de Programação | **Projeto 1** |

**RELATÓRIO DO PROJETO 1**

**CALCULADORA PARA COMPLEXOS EM NOTAÇÃO POLACA INVERSA**

**Trabalho realizado por:** Ana Claro (2020223041) e Margarida Rodrigues (2020223166), Grupo 1.

**Especificação**

Dados números complexos (constituídos por uma parte real e uma parte imaginaria) e respetivos operadores em Notação Polaca Inversa, calcula o resultado da expressão.

**Entradas**

Números reais que correspondem à parte imaginária e real de um complexo e operadores;

**Saída**

Resultado da expressão introduzida, tendo em conta todos os números e operadores.

**Utilização**

O programa começa por pedir ao utilizador para introduzir uma expressão em NPI e em seguida um ‘.’ para simbolizar o término da expressão. De seguida, o programa calcula o valor da expressão e imprime-o no ecrã.

Por exemplo, se o utilizador quiser inserir o complexo “3+4i” deverá escrever “3 4 .” .

**Algoritmo**

|  |  |
| --- | --- |
| 1 | ler (aux) |
| 2 | se (aux= ‘.’) então |
| 3 | Pilhavazia |
| 4 | senao |
| 5 | enquanto (aux !=0) |
| 6 | se (aux=operador) |
| 7 | op2 = top (p1); |
| 8 | pop(&p1); |
| 9 | op1 = top(p1); |
| 10 | pop(&p1); |
| 11 | se (aux=+) |
| 12 | somaComplexos |
| 13 | se (aux=-) |
| 14 | subtraiComplexo |
| 15 | se (aux=\*) |
| 16 | multiplicaComplexo |
| 17 | se (aux=/) |
| 18 | divideComplexos |
| 19 | push(\*res, &p1) |
| 20 | se não |
| 21 | operando.re = atoi(aux) |
| 22 | scanf("%s",aux) |
| 23 | operando.im=atoi(aux) |
| 24 | push(operando, &p1) |
| 25 |  |
| 26 | scanf("%s", aux) |
| 27 | \*res = top(p1) |
| 28 | EscreveComplexo |

A função malloc aloca o espaço necessário para um dado valor.

As funções top, pop e push, relativamente a uma pilha, desempenham as funções de:

- top: devolve o elemento localizado no topo da pilha;

- pop: remove o elemento localizado no topo da pilha;

- push: introduz um novo elemento no topo da pilha.

As funções relacionadas com operadores, nomeadamente *somaComplexos, subtraiComplexos, divideComplexos* e *multiplicaComplexos*, dizem respeito a operações elementares realizadas entre dois números complexos.

A função *ler*, guarda em formato de pilha números complexos e operadores. A função *EscreveComplexo*, imprime no ecrã um numero complexo na forma a+bi.

**Estruturas de Dados**

Complexos e pilha.

**Particularidades**

As operações são feitas a pares….

Após cada expressão deverá ser introduzido um “.” para identificar o fim da expressão .

**Exemplos (apresentados em Linux)**

1. **Uma imagem com texto

   Descrição gerada automaticamenteSoma de Complexos:**
2. **Uma imagem com texto

   Descrição gerada automaticamenteSubtração de Complexos:**
3. **Produto de Complexos:**

**Uma imagem com texto

Descrição gerada automaticamente**

1. **Divisão entre Complexos:**

**Uma imagem com texto

Descrição gerada automaticamente**

1. **Várias operações:**

**- Exemplo 5.1:**

**Uma imagem com texto

Descrição gerada automaticamente**

**- Exemplo 5.2:**

**Uma imagem com texto

Descrição gerada automaticamente**

1. **Uma imagem com texto

   Descrição gerada automaticamenteDivisão de um complexo por 0+0i:**