



Calculadora RPN: Calculando Expressões Matemáticas de Forma Eficiente

A calculadora RPN (Notação Polonesa Inversa) é uma ferramenta poderosa e eficiente para realizar cálculos matemáticos. Ao invés de usar a tradicional entrada de operadores e operandos, a RPN usa uma abordagem de pilha, tornando os cálculos mais diretos e intuitivos.

Participantes do grupo:

Ágata Cesquim

Bruno Martinho

Enzo Dorta

Giovanni de Pita

Murilo Enrico

Vicente Venancio

Estrutura da Pilha

Itens da Pilha

A pilha armazena os operandos (valores) à medida que a expressão é avaliada.

Topo da Pilha

O topo da pilha é onde os novos operandos são adicionados e os operadores são aplicados.

Estrutura da Pilha

```
typedef struct {  
    double itens[TAMANHO_MAXIMO];  
    int topo;  
} Pilha;
```

Processando Expressões RPN

Leitura dos Tokens

A expressão RPN é lida token por token, onde cada token pode ser um operando (número) ou um operador.

1

Empilhamento de Operandos

Os operandos são empilhados conforme são encontrados na expressão.

2

Aplicação de Operadores

Quando um operador é encontrado, os operandos necessários são retirados da pilha, a operação é realizada e o resultado é empilhado.

3

Interação com o Usuário

Entrada de Dados

O programa deve solicitar ao usuário que digite a expressão RPN e tratá-la adequadamente.

Exibição de Resultados

Após o cálculo, o programa deve exibir o resultado de forma clara e legível.

Opção de Continuação

O programa deve oferecer a opção de o usuário continuar realizando novos cálculos.



Vantagens da Calculadora RPN

1

Simplicidade

A abordagem de pilha da RPN simplifica a entrada e o processamento de expressões.

2

Eficiência

A RPN elimina a necessidade de lidar com a precedência dos operadores, tornando os cálculos mais rápidos.

3

Facilidade de Uso

Após um breve período de adaptação, a RPN se torna intuitiva e fácil de usar.

Como funciona a HP12c ?

USAREMOS O EXMPLO QUE O USUÁRIO UTILIZARÁ A OPERAÇÃO : 10 10 +

```
Digite a expressão RPN: 10 10 +  
Fórmula Algébrica: 10 10 +  
Resultado: 20.00  
Deseja continuar (S/N)?
```


Iniciação da Pilha

```
63  int main() {  
64      Pilha pilha;  
65      pilha.topo = -1;  
66  
67      char expressao[TAMANHO_MAXIMO];  
68      char continuar;
```

Inserção da formula

```
do {  
    printf("Digite a expressão RPN: ");  
    fgets(expressao, TAMANHO_MAXIMO, stdin);  
    expressao[strlen(expressao, "\n")] = '\0';
```

Inspeção e demonstração da operação

```
printf("Fórmula Algébrica: ");
for (int i = 0; expressao[i] != '\0'; i++) {
    if ((expressao[i] >= '0' && expressao[i] <= '9') || expressao[i] == ' ' ||
        expressao[i] == '.') {
        printf("%c", expressao[i]);
    } else {
        printf(" %c", expressao[i]);
    }
}
printf("\n");
```


Chamada da função

```
86 double resultado = calcularRPN(expressao);  
87 printf("Resultado: %.2f\n", resultado);
```

Início da função

```
21 double calcularRPN(char *expressao) {  
22     Pilha pilha;  
23     pilha.topo = -1;  
24  
25     double operando1, operando2;  
26     char *token = strtok(expressao, " ");  
27
```

Definindo a operação

```
28 while (token != NULL) {
29     if (isdigit(*token)) {
30         empilhar(&pilha, atof(token));
31     } else {
32         operando2 = desempilhar(&pilha);
33         operando1 = desempilhar(&pilha);
34
35         switch (*token) {
36             case '+':
37                 empilhar(&pilha, operando1 + operando2);
38                 break;
39             case '-':
40                 empilhar(&pilha, operando1 - operando2);
41                 break;
42             case '*':
43                 empilhar(&pilha, operando1 * operando2);
44                 break;
45             case '/':
46                 if (operando2 == 0) {
47                     printf("Erro: Divisão por zero\n");
48                     exit(EXIT_FAILURE);
49                 }
50                 empilhar(&pilha, operando1 / operando2);
51                 break;
52             default:
53                 printf("Erro: Operador inválido\n");
54                 exit(EXIT_FAILURE);
55         }
56     }
57     token = strtok(NULL, " ");
58 }
```


Mostrando o resultado

```
86     double resultado = calcularRPN(expressao);  
87     printf("Resultado: %.2f\n", resultado);
```

```
Digite a expressão RPN: 10 10 +  
Fórmula Algébrica: 10 10 +  
Resultado: 20.00  
Deseja continuar (S/N)?
```