

$$6) (A+B) * D + E / (F + A * D) + C$$

Pilha  
 $\boxed{C}$

Saída

---

 $\boxed{C}$ 

A

---

 $\boxed{+}$   
 $\boxed{C}$ 

AB

---

 $\boxed{+}$ 

AB +

---

 $\boxed{*}$ 

AB +

---

 $\boxed{*}$ 

AB + D

$$\boxed{\phantom{+}} \quad AB + D *$$

---

$$\boxed{+} \quad AB + D *$$

---

$$\boxed{+} \quad AB + D * E$$

---

$$\boxed{\begin{array}{c} / \\ + \end{array}} \quad AB + D * E$$

---

$$\boxed{\begin{array}{c} C \\ / \\ + \end{array}} \quad AB + D * E$$

---

$$\boxed{\begin{array}{c} C \\ / \\ + \end{array}} \quad AB + D * E F$$

---

+

(

/

+

$$AB + D * EF$$

---

+

(

/

+

$$AB + D * EFA$$

---

\*

+

(

/

+

$$AB + D * EFA$$

---

$\left[ \begin{array}{c} * \\ + \\ < \\ / \\ + \end{array} \right]$

$AB + D * EFAD$

---

$\left[ \begin{array}{c} / \\ + \end{array} \right]$

$AB + D * EFAD * +$

---

$\left[ + \right]$

$AB + D * EFAD * + /$

---

$\left[ \right]$

$AB + D * EFAD * + / +$

---

$\left[ + \right]$

$AB + D * EFAD * + / +$

---

$\boxed{+} AB + D * EFAD * + / + C$

---

$\boxed{+} AB + D * EFAD * + / + C +$

---

```

int quatrocores(int **mat, int n, int *cores){
    /*
    assumindo que as posições de cores estão preenchidas com 1
    assumindo que cores[i] representa a cor do i-ésimo pais
    assumindo que cada cor é indicada por um número inteiro, tq 1 e 2 são cores distintas
    */
    pilha *decisao = criaPilha(n+1);
    int pais, vizinho, livre, i;

    for(i=0;i<n;i++){
        cores[i] = 1;
    }

    pais = 0;
    livre = 0;
    while(pais < n){
        /*Enquanto não houver verificado os n-1 paises*/
        livre = 0;
        while(!livre && cores[pais] <= 4 ){ /*Enquanto não estiver livre e cor <=4 */
            for(i = 0; i < pais; i++){
                if(mat[pais][i] == 1 && cores[i] == cores[pais] && pais != i){
                    /*Se for vizinho, com cor igual e o vizinho não for o próprio pais*/
                    livre = 0;
                    break;
                }
                else{
                    /*a cor atual não é usada por algum vizinho*/
                    livre = 1;
                }
            }
            if(pais == 0){
                /*exceção para o primeiro pais*/
                livre = 1;
            }
            if(livre == 0){
                /*Se não estiver livre, atribui uma cor nova*/
                cores[pais]++;
            }
        }
        if(livre){
            /*Se livre, avança para o próximo pais e empilha a cor do pais atual*/
            pais++;
            empilha(decisao, cores[pais-1]);
        }
        else{
            /*Backtracking*/
            if(pilhaVazia(decisao)){
                destroiPilha(decisao);
                return 0;
            }
            else{
                pais--;
                /*
                *Retorna para o pais anterior, muda a cor dele para a empilhada e refaz o processo*/
                cores[pais] = desempilha(decisao);
                cores[pais]++;
            }
        }
    }
}

```

```
    }  
    destroiPilha(decisao);  
    return 1;  
}
```