## 2．ソースコードとその説明

```
#include <stdio.h>
#define N 50

/*スタックのpush*/
void push(int x, int S[]){
        if(S[0]>N-1){ /*例外処理（スタックが満タン）*/
                printf("スタックにこれ以上要素は入りません¥n");
        }
        else            /*S[0]にはSに格納されている数字の数を格納し、S[0]番目(一番うしろ)に新たな値を格納*/
                S[++S[0]] = x;
}

/*スタックのpop*/
int pop(int S[]){
        if(S[0]==0){ /*例外処理（スタックが空）*/
                printf("スタックは空です。要素はありません。¥n");
                return 0;
        }
        else /*S[0]番目（１番後ろ）に格納されている数を返し、S[0]をひとつ減らす*/
                return S[S[0]--];
}

/*足し算関数*/
void add(int S[]){
        int a,b; /*変数の宣言*/

        a = pop(S); /*Sの１番後ろの値を取り出し,aに代入*/
        b = pop(S); /*Sの１番後ろの値を取り出し,bに代入*/

        push(a+b, S); /*a+bをSの一番後ろに格納*/
}
```

```
/*掛け算関数*/
void mul(int S[]){
        int a,b; /*変数の宣言*/

        a = pop(S); /*Sの１番後ろの値を取り出し,aに代入*/
        b = pop(S); /*Sの１番後ろの値を取り出し,bに代入*/

        push(a*b,S); /*a*bをSの一番後ろに格納*/
}

/*char型からint型への変換関数*/
int c_i(char x){

        x = x - 48; /*char型の数字をint型の数字にするため-48*/

        return x;
}

/*逆ポーランド記法の計算を実行する関数*/
void re_p(char A[],int S[],int length){

        int i,x;  /*変数の宣言*/

        for(i=0;i<length;i++){ /*A[0]～A[length-1]まで順に見ていく*/
                if(A[i]==43) /*要素が+ であるとき*/
                        add(S);  /*関数addにSをわたす*/
                else if(A[i]==42) /*要素が* であるとき*/
                        mul(S);  /*関数mulにSをわたす*/
                else          /*要素が数字であるとき*/
                        push(c_i(A[i]),S); /*int型に変換してからSの一番後ろに格納す
る*/
        }
        printf("%d",S[1]); /*計算結果の表示*/
}
```

```
/*メイン関数　データの入出力*/
int main(){
        int S[N]; /*変数の宣言*/
        char A[N];
        int i = 0; /*初期化*/
        int length = 0;

        S[0] = 0;

        gets(A);  /*配列Aにキーボードから文字列を入力*/
        while(A[i]!=0) /*入力された文字列の長さを求める*/
                i++;
        length = i;

        re_p(A, S, length); /*配列Aとスタックに利用する配列SとAの長さを関数re_pに渡す*/

        return 0;
}
```