

① STACK

pseudocode:

void push (int n)

{

if (top == (Size - 1))

{

printf("Overflow");

}

else {

top++;

Stack[top] = n;

}

}

void pop ()

{

if (top == -1)

printf("Underflow");

}

else {

printf("%d", Stack[top]);

top--;

}

}


```
void display ( )
```

```
{
```

```
    if (top == -1) {
```

```
        printf("Stack empty");
```

```
    } else {
```

```
        for (top; top > 0; top--)
```

```
        {
```

```
            printf("%d \t", stack[top]);
```

```
        }
```

```
    }
```

```
}
```

01/01/24

Output:

1. push 2. pop 3. display 4. exit

2

Stack underflow

1. push 2. pop 3. display 4. exit

1.

enter value: 1

1. push 2. pop 3. display 4. exit

1

enter value: 2

1. push 2. pop 3. display 4. exit

1

enter value: 3

1. push 2. pop 3. display 4. exit

1

enter value: 4

Stack overflow

② ~~Postfix~~ evaluation

```

void main() {
    char *e, char exp[20];
    scanf("%s", exp);
    e = exp;

    while (*e != '\0')
    {
        if (isdigit(*e))
        {
            push(*e - 48);
        }
        else
        {
            n1 = pop();
            n2 = pop();
            switch(*e)
            {
                case '+': n3 =

```

1. push 2. pop 3. display 4. exit

2. deleted value is 3

1. push 2. pop 3. display 4. exit

3

1 2

1. push 2. pop 3. display 4. exit

4

② infix to postfix

```
int priority(char x)
{
    if (x == '(')
        return 0;
    if (x == '+' || x == '-')
        return 1;
    if (x == '*' || x == '/')
        return 2;
    return 0;
}
```

```
int main ()
{
```

```
    char exp[100], *e, x;
    scanf("%s", exp);
    e = exp;
```

```
    while (*e != '\0')
```

```
    {
```

```
        if (isalnum(*e))
```

```
            printf("x.c", *e);
```

```
        else if (*e == '(')
```

```
            push(*e);
```

```
        else if (*e == ')')
```

```
            { x = pop();
```

```
              while (x != '(')
```

```
              {
```

```
                  if (pop())
```

```
                      printf("x.c", x);
```

```
                  x = pop();
```


else

{

while (Priority(Stack[top]) >= Priority(*e))

{ printf("%c", pop()); }

push(*e);

}

e++;

}

~~Sum~~
~~11/12/24~~

Output:

enter expression: a+b*c-d+h*i

abc*+d-hi*+