Dynamically create a 2D array using single pointer

- Input:
  - *N*: number of students in a class (Number of columns)
  - *M*: number of examinations of each student (Number of rows)

number of students

number of examinations

```
/* Declare a variable*/
int *grade;
/* Allocate memory of size M x N */
MALLOC(grade, M * N * sizeof (int));
```

*The address of A[i][j] is base_address+ (nRows*j + i)*size*

Assign score 80 to exam [1] for student [2]

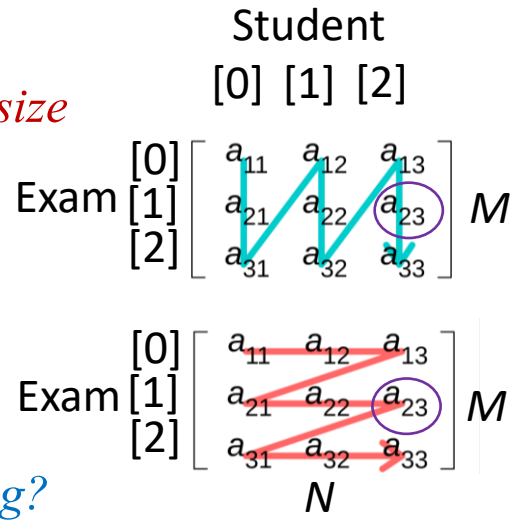
Column major

```
grade[ M*2+1 ]=80;
```

N*1+2 or N+2

Row major

```
grade[    Q1     ]=80;
```

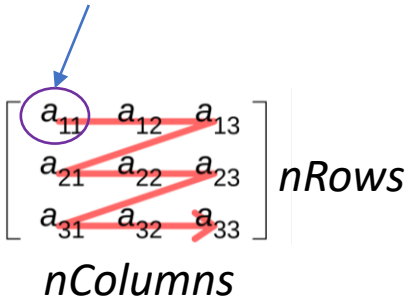


Q2: What is the address of A[i][j] in row-major mapping?

*base_address + (nColumns*i + j)*size*

- Q2: Let **A** be a two-dimensional array with *nRows* rows and *nColumns* columns. Let *S* be the size of each element in **A** and *alpha* be the base address of **A**. When the array **A** is stored using row-major order, what is the **address** of an arbitrary element *A*[*i*][*j*] ?

The address of A[0][0] is base address.



$$base\_address + (nColumns*i + j)*size$$

$$alpha + (nColumns*i + j)*S$$

Some answers needed to be modified:

(N*j)+i	A[i*N+j]	A + (nRows * i + j) * S	alpha+(n*j+i)*S

- Steps:
  - Scan the expression from left to right
  - When a left bracket, “[”, “{”, or “(”, is encountered, push it into stack.
  - When a right bracket, “]”, “}”, or “)”, is encountered, pop the top element from stack and check whether they are matched.
    - If the right bracket and the pop out element are not matched, return False.
  - After scanning the whole expression, if the stack is not empty, return False.

- Examine whether the brackets in the following expression are balanced.

{[(a+b)*c+d-e]/(f+g)-(h+j)*k-l)}/(m-n)

  - Q3: Are the brackets in the expression balanced?
  - Q4: How many times of push and pop are required?
  - Q5: Please list the order of push and pop.

{[(a+b)*c+d-e]/(f+g)-(h+j)*k-l)}/(m-n)

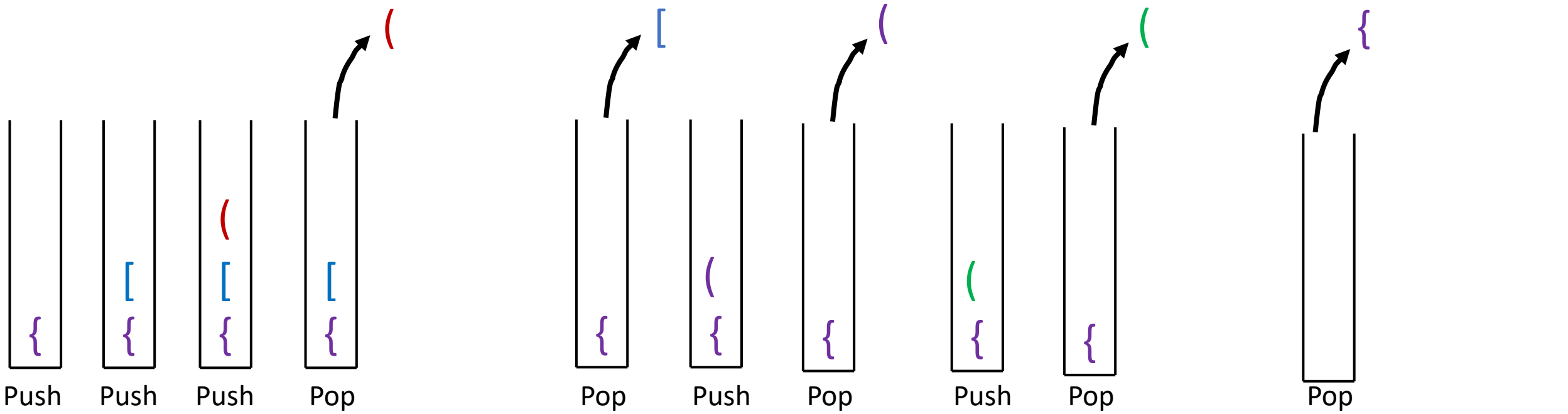
"(" and ")" are Matched. Continue scanning.

"[" and "]" are Matched. Continue scanning.

"(" and ")" are Matched. Continue scanning.

"(" and ")" are Matched. Continue scanning.

"{" and ")" are not matched. **Return False.**



Note:

1. If the pop element and the right bracket are not matched, false will be returned (the program is **terminated**).
2. If a right bracket is encountered, pop an element from stack and then check whether they are matched.

1 pop and 2 pushes for "}/(m-n)"
-> won't happen