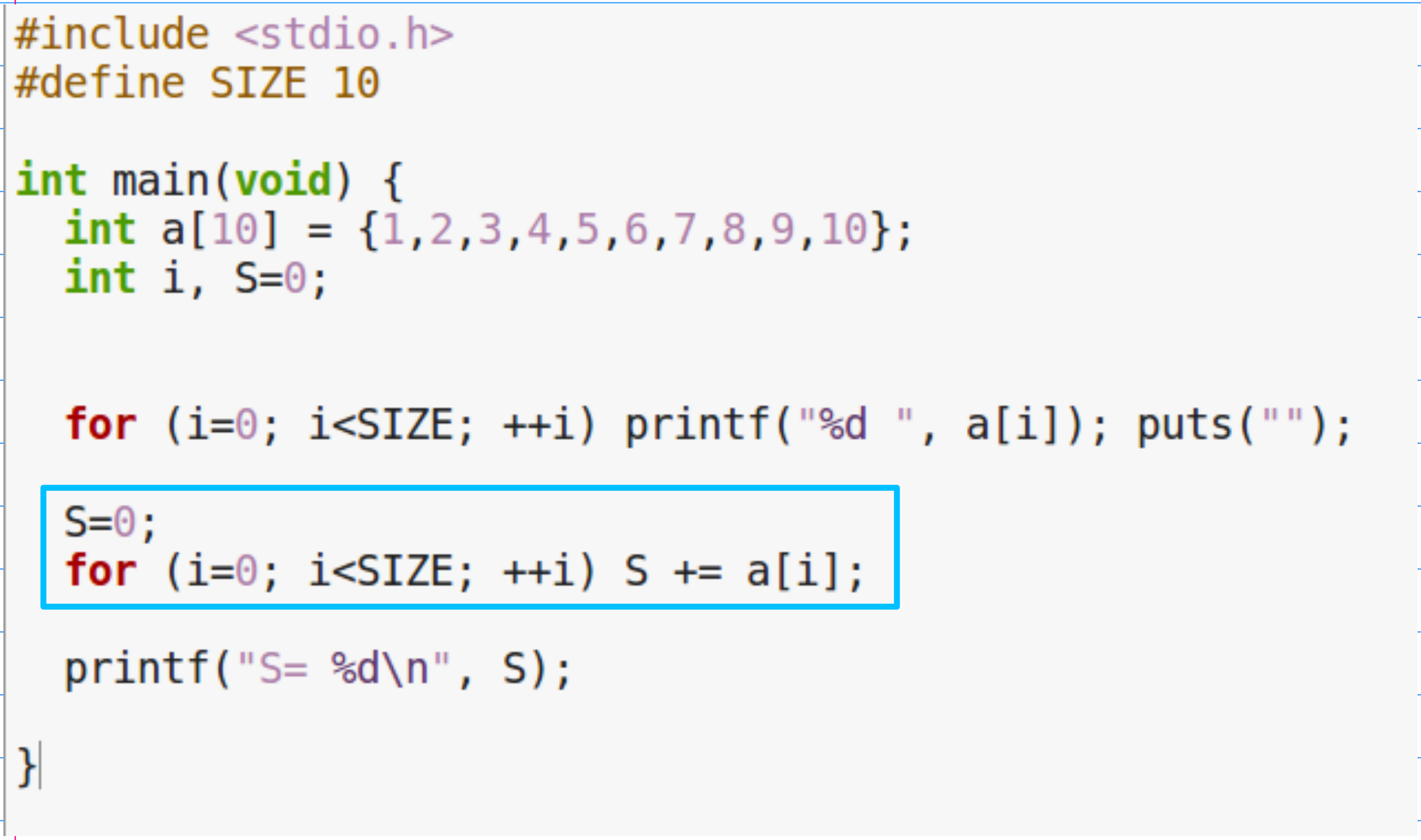
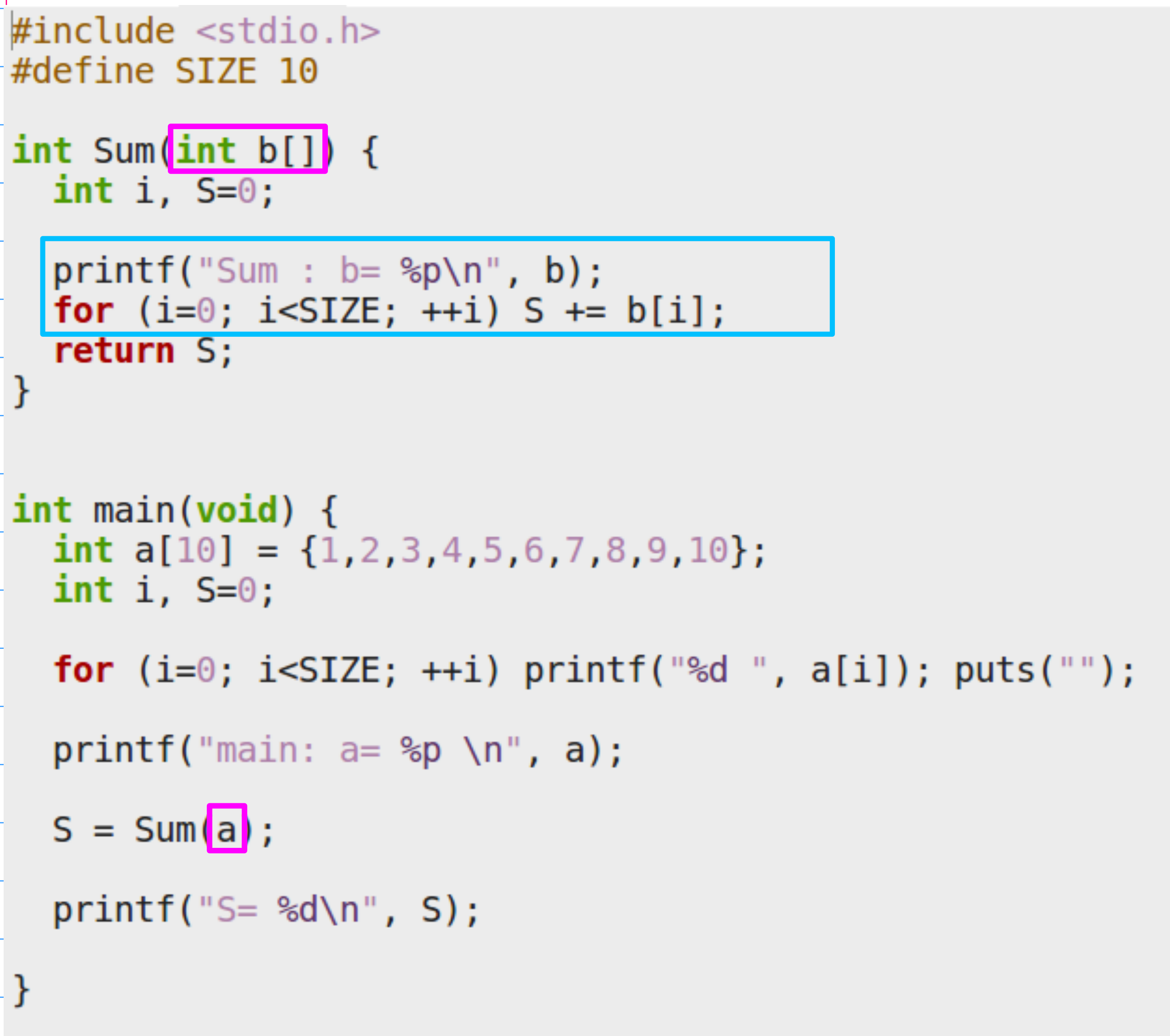
C Programming
Day11.B

2017.10.13

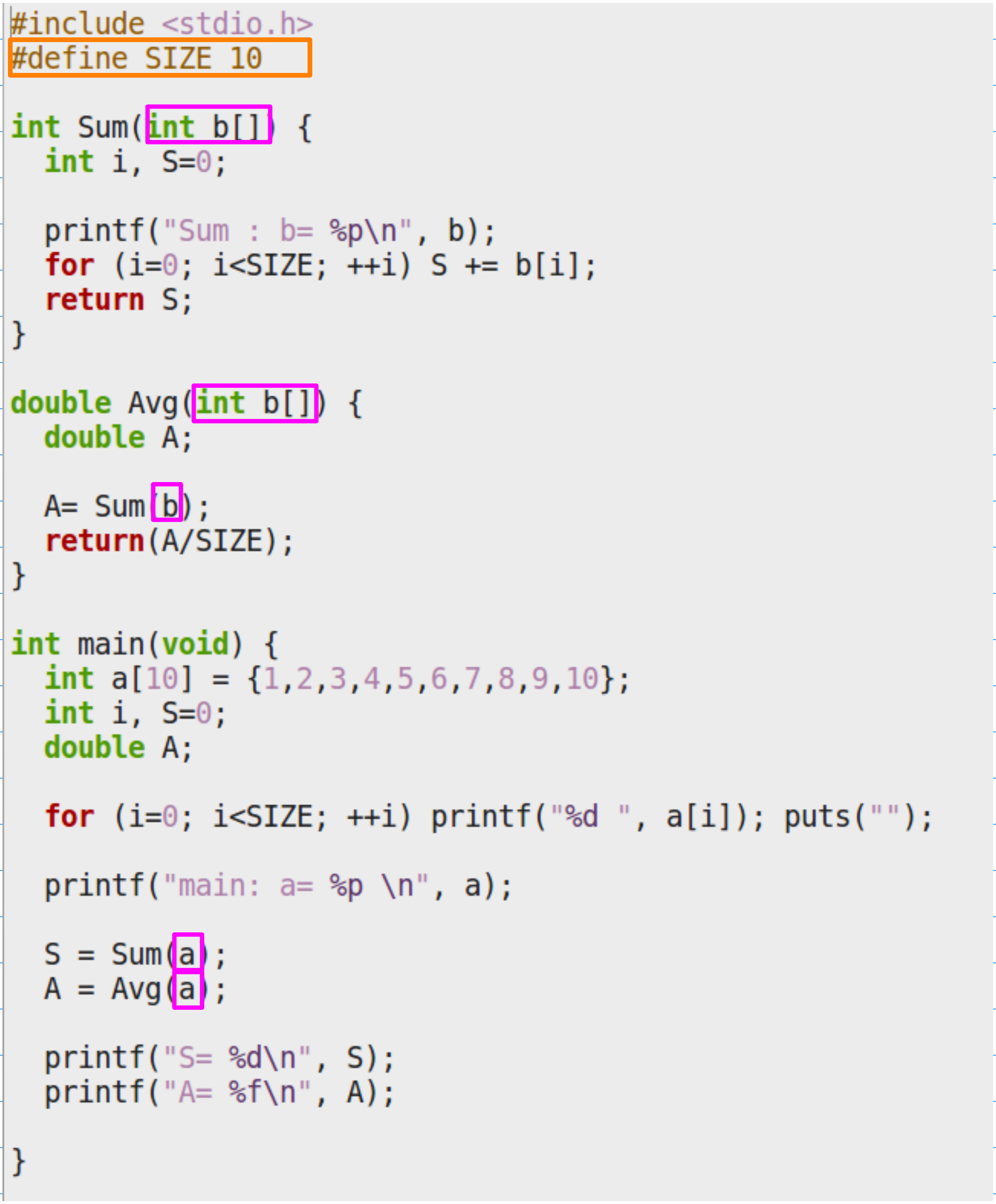
Arrays, Mulit-dimension,



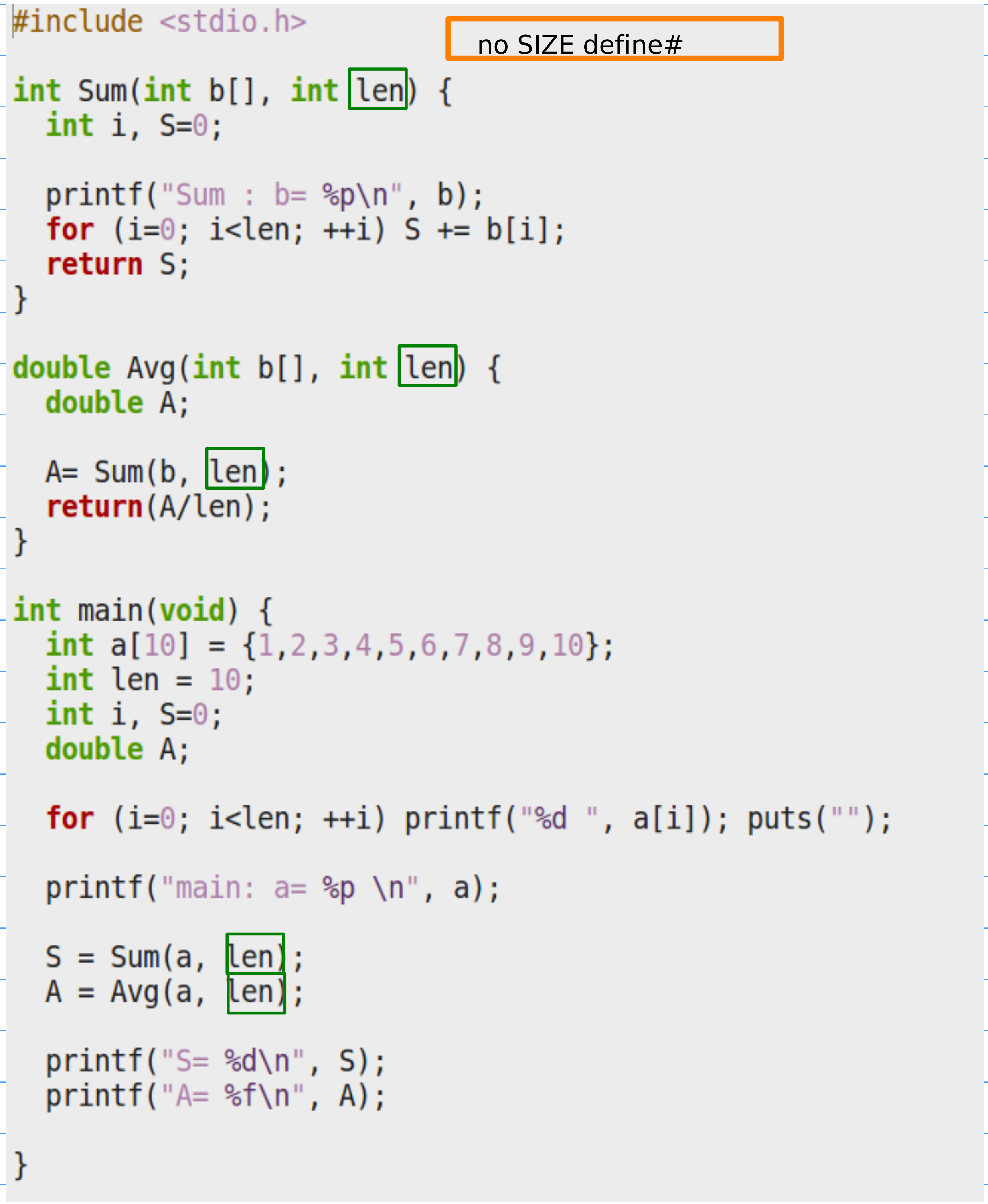
```
#include <stdio.h>
#define SIZE 10

int main(void) {
  int a[10] = {1,2,3,4,5,6,7,8,9,10};
  int i, S=0;


  for (i=0; i<SIZE; ++i) printf("%d ", a[i]); puts("");

  S=0;
  for (i=0; i<SIZE; ++i) S += a[i];

  printf("S= %d\n", S);

}
```



```c
#include <stdio.h>
#define SIZE 10

int Sum(int b[]) {
  int i, S=0;

  printf("Sum : b= %p\n", b);
  for (i=0; i<SIZE; ++i) S += b[i];
  return S;
}


int main(void) {
  int a[10] = {1,2,3,4,5,6,7,8,9,10};
  int i, S=0;

  for (i=0; i<SIZE; ++i) printf("%d ", a[i]); puts("");

  printf("main: a= %p \n", a);

  S = Sum(a);

  printf("S= %d\n", S);

}
```

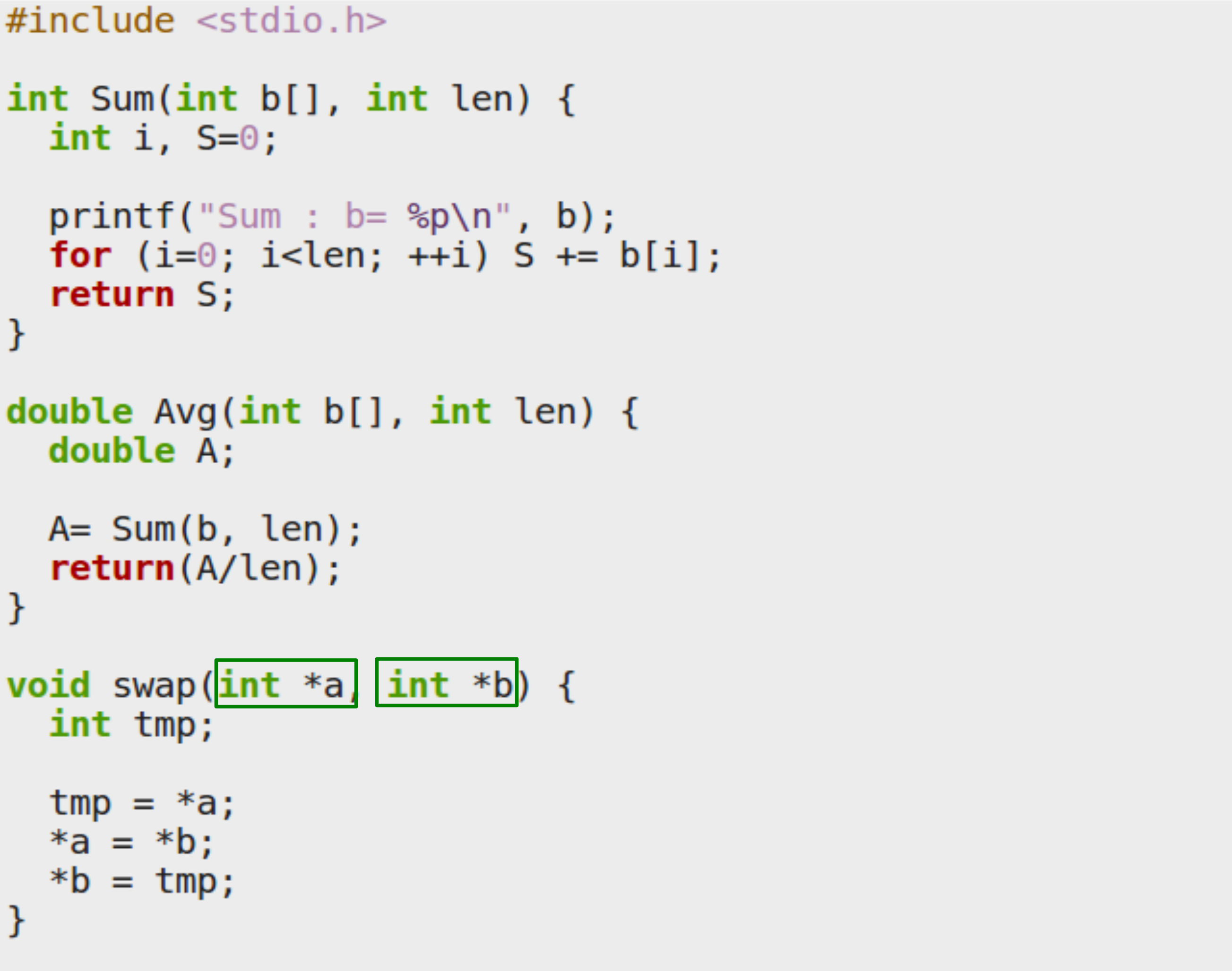


```c
#include <stdio.h>
#define SIZE 10

int Sum(int b[]) {
  int i, S=0;

  printf("Sum : b= %p\n", b);
  for (i=0; i<SIZE; ++i) S += b[i];
  return S;
}

double Avg(int b[]) {
  double A;

  A= Sum(b);
  return(A/SIZE);
}

int main(void) {
  int a[10] = {1,2,3,4,5,6,7,8,9,10};
  int i, S=0;
  double A;

  for (i=0; i<SIZE; ++i) printf("%d ", a[i]); puts("");

  printf("main: a= %p \n", a);

  S = Sum(a);
  A = Avg(a);

  printf("S= %d\n", S);
  printf("A= %f\n", A);

}
```

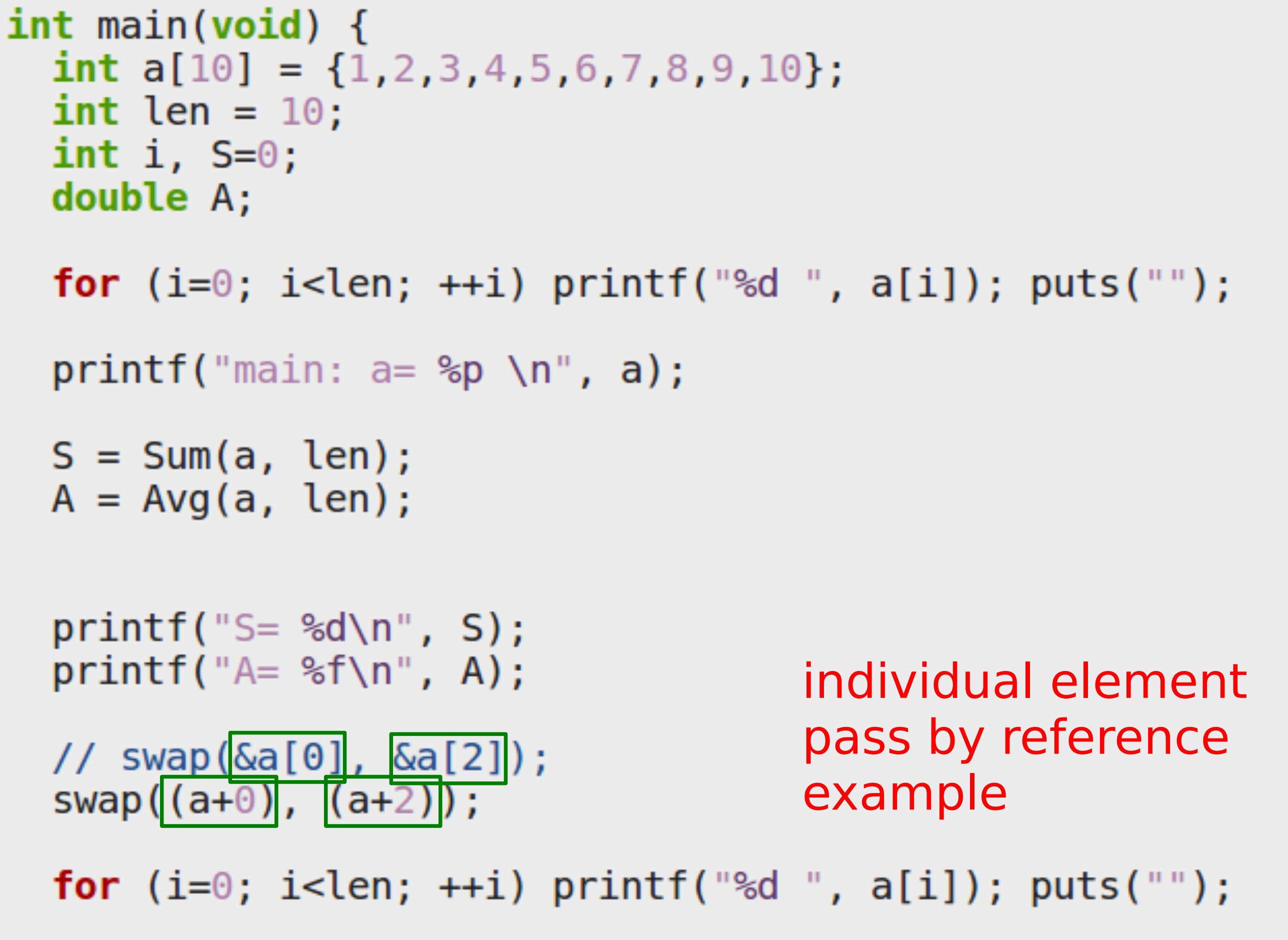


no SIZE define#

```
#include <stdio.h>

int Sum(int b[], int len) {
  int i, S=0;

  printf("Sum : b= %p\n", b);
  for (i=0; i<len; ++i) S += b[i];
  return S;
}

double Avg(int b[], int len) {
  double A;

  A= Sum(b, len);
  return(A/len);
}

int main(void) {
  int a[10] = {1,2,3,4,5,6,7,8,9,10};
  int len = 10;
  int i, S=0;
  double A;

  for (i=0; i<len; ++i) printf("%d ", a[i]); puts("");

  printf("main: a= %p \n", a);

  S = Sum(a, len);
  A = Avg(a, len);

  printf("S= %d\n", S);
  printf("A= %f\n", A);

}
```

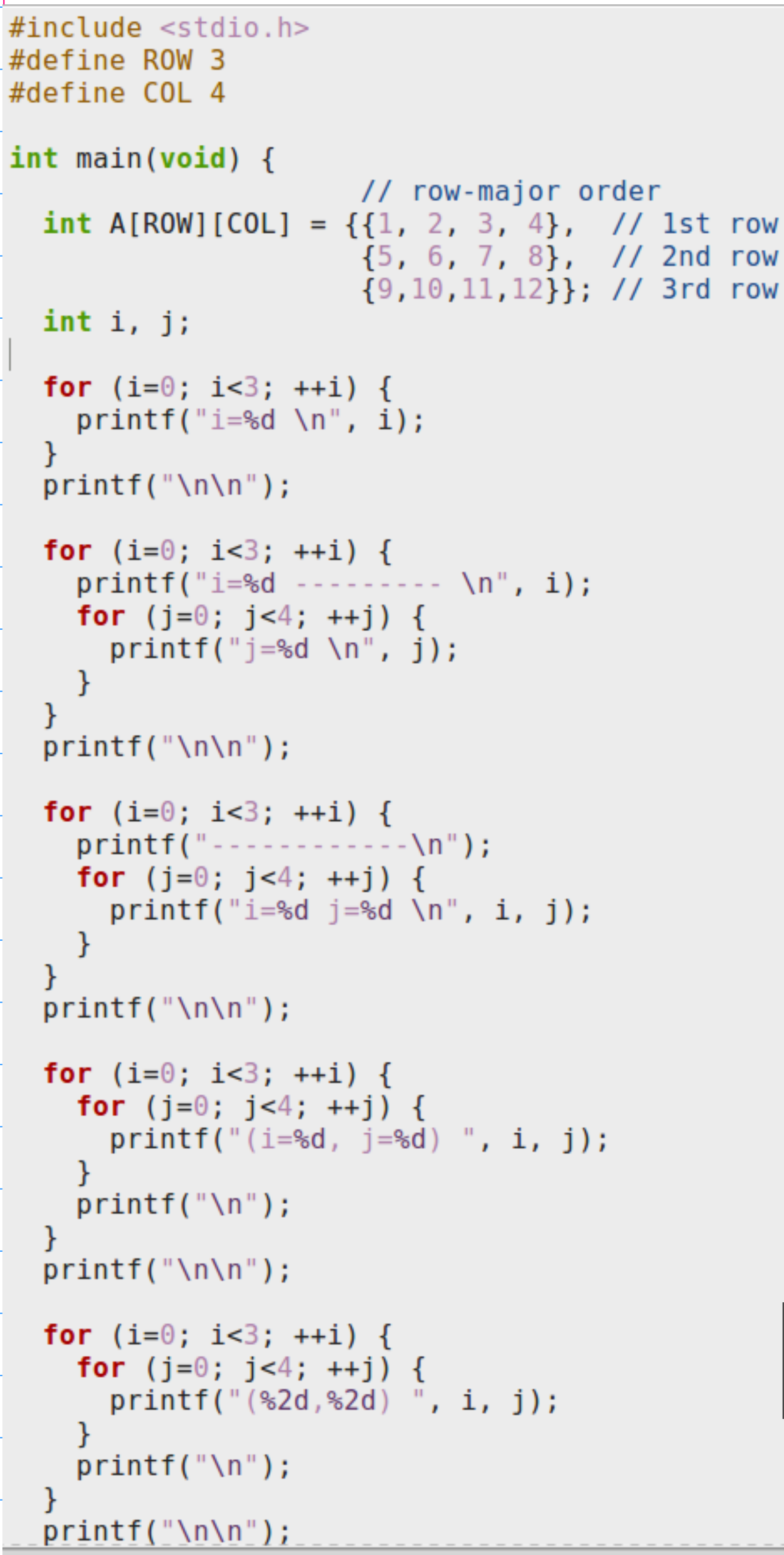


```c
#include <stdio.h>

int Sum(int b[], int len) {
  int i, S=0;

  printf("Sum : b= %p\n", b);
  for (i=0; i<len; ++i) S += b[i];
  return S;
}

double Avg(int b[], int len) {
  double A;

  A= Sum(b, len);
  return(A/len);
}

void swap(int *a, int *b) {
  int tmp;

  tmp = *a;
  *a = *b;
  *b = tmp;
}
```

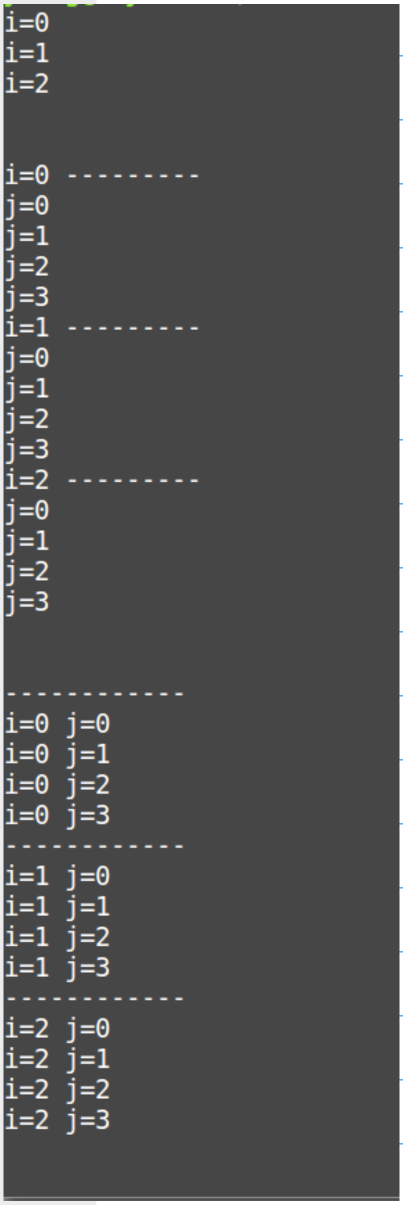


```c
int main(void) {
  int a[10] = {1,2,3,4,5,6,7,8,9,10};
  int len = 10;
  int i, S=0;
  double A;

  for (i=0; i<len; ++i) printf("%d ", a[i]); puts("");

  printf("main: a= %p \n", a);

  S = Sum(a, len);
  A = Avg(a, len);


  printf("S= %d\n", S);
  printf("A= %f\n", A);

  // swap(&a[0], &a[2]);
  swap((a+0), (a+2));

  for (i=0; i<len; ++i) printf("%d ", a[i]); puts("");
```

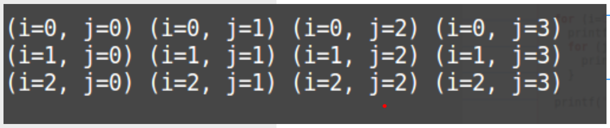
individual element
pass by reference
example



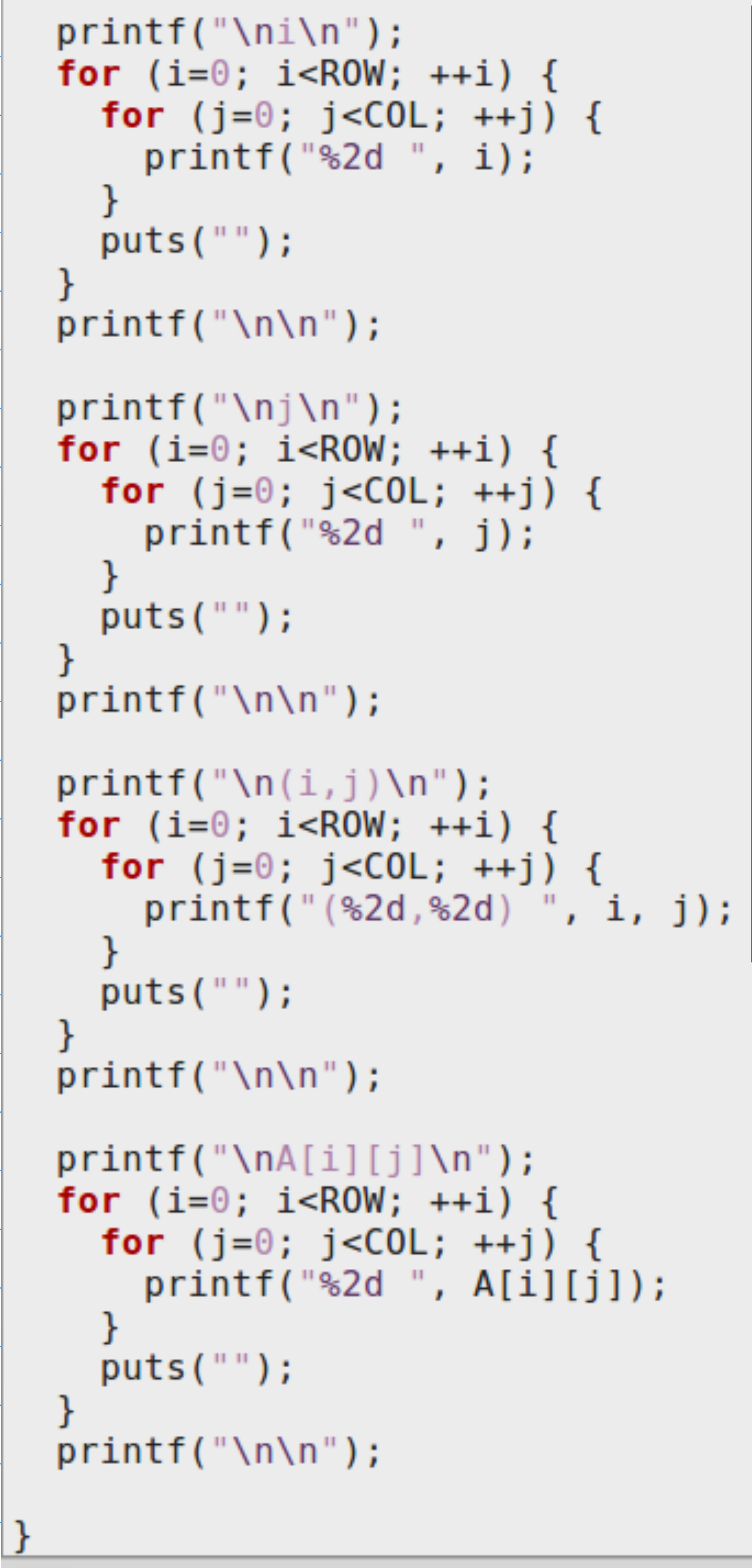
```c
#include <stdio.h>
#define ROW 3
#define COL 4

int main(void) {
                     // row-major order
  int A[ROW][COL] = {{1, 2, 3, 4},  // 1st row
                     {5, 6, 7, 8},  // 2nd row
                     {9,10,11,12}}; // 3rd row
  int i, j;

  for (i=0; i<3; ++i) {
    printf("i=%d \n", i);
  }
  printf("\n\n");

  for (i=0; i<3; ++i) {
    printf("i=%d --------- \n", i);
    for (j=0; j<4; ++j) {
      printf("j=%d \n", j);
    }
  }
  printf("\n\n");

  for (i=0; i<3; ++i) {
    printf("------------\n");
    for (j=0; j<4; ++j) {
      printf("i=%d j=%d \n", i, j);
    }
  }
  printf("\n\n");

  for (i=0; i<3; ++i) {
    for (j=0; j<4; ++j) {
      printf("(i=%d, j=%d) ", i, j);
    }
    printf("\n");
  }
  printf("\n\n");

  for (i=0; i<3; ++i) {
    for (j=0; j<4; ++j) {
      printf("(%2d,%2d) ", i, j);
    }
    printf("\n");
  }
  printf("\n\n");
```



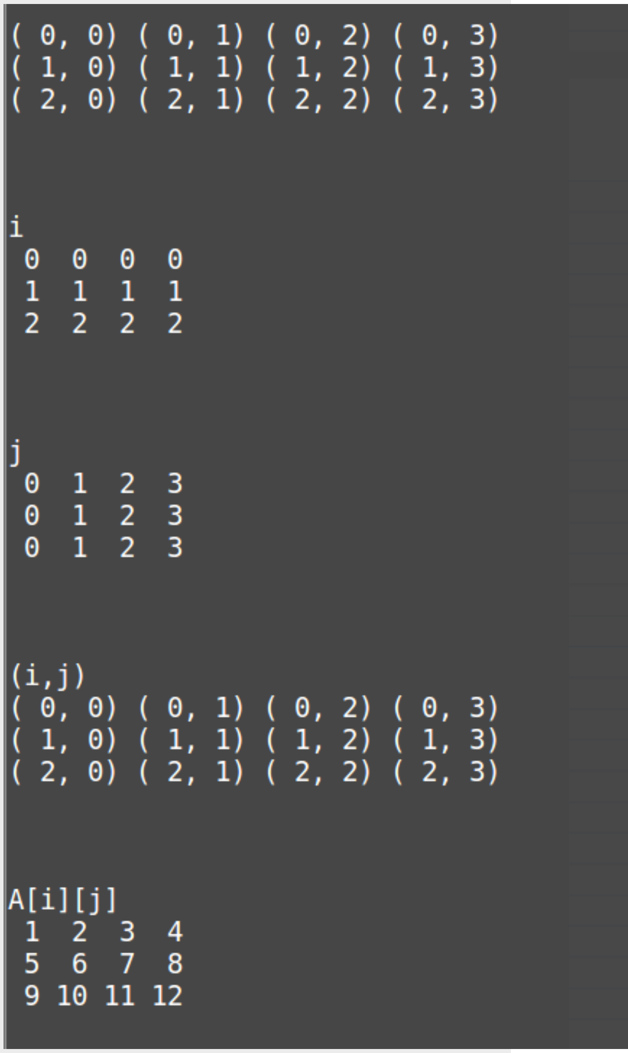
```
i=0
i=1
i=2


i=0 ---------
j=0
j=1
j=2
j=3
i=1 ---------
j=0
j=1
j=2
j=3
i=2 ---------
j=0
j=1
j=2
j=3


------------
i=0 j=0
i=0 j=1
i=0 j=2
i=0 j=3
------------
i=1 j=0
i=1 j=1
i=1 j=2
i=1 j=3
------------
i=2 j=0
i=2 j=1
i=2 j=2
i=2 j=3
```



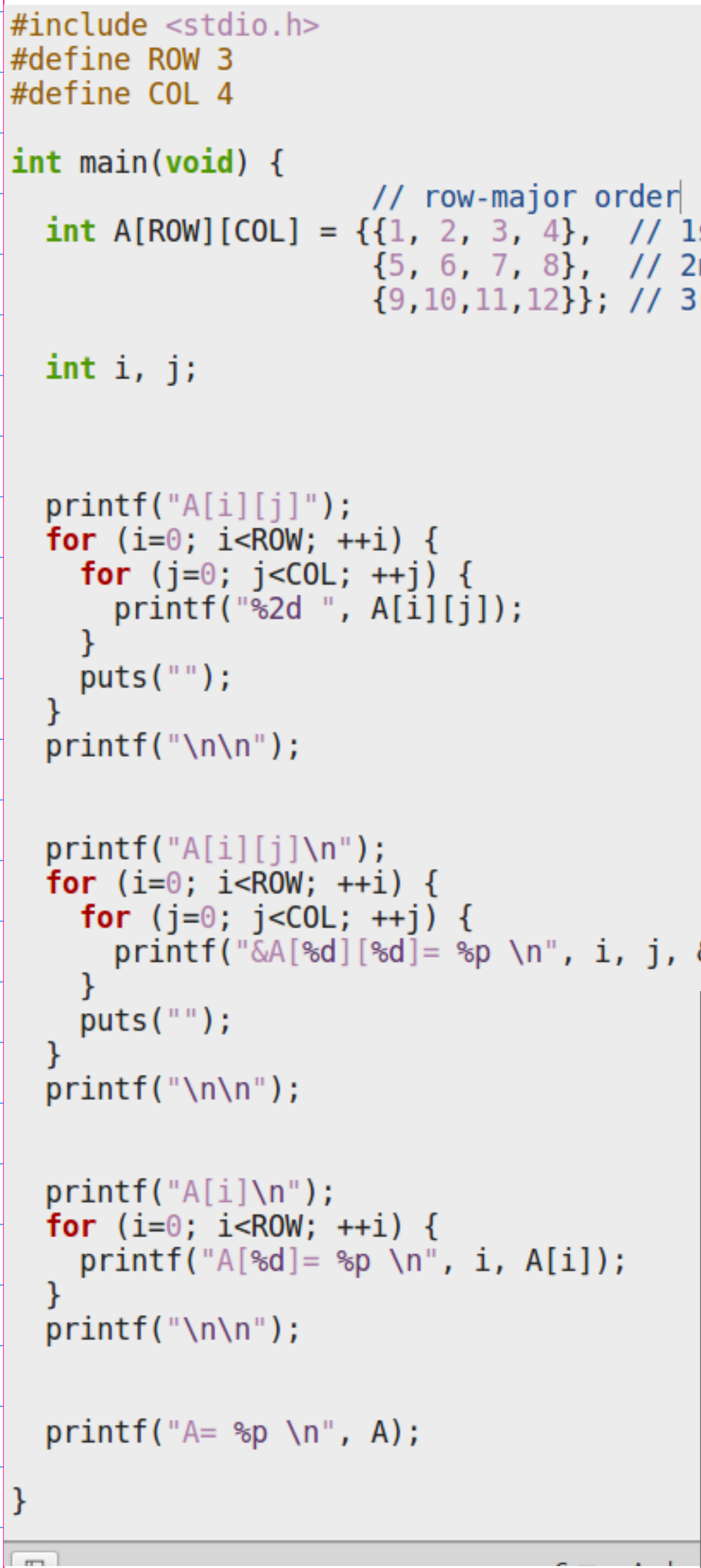
```
(i=0, j=0) (i=0, j=1) (i=0, j=2) (i=0, j=3)
(i=1, j=0) (i=1, j=1) (i=1, j=2) (i=1, j=3)
(i=2, j=0) (i=2, j=1) (i=2, j=2) (i=2, j=3)
```



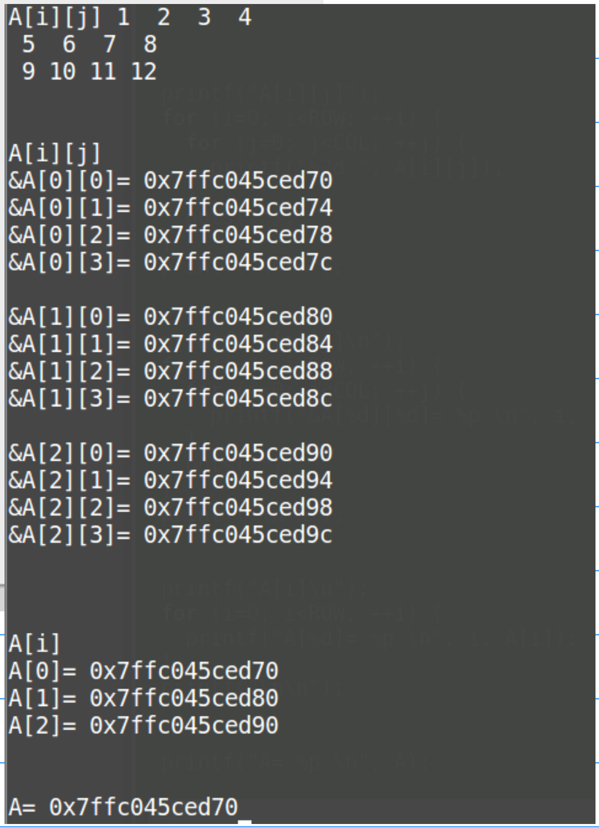
```c
  printf("\ni\n");
  for (i=0; i<ROW; ++i) {
    for (j=0; j<COL; ++j) {
      printf("%2d ", i);
    }
    puts("");
  }
  printf("\n\n");

  printf("\nj\n");
  for (i=0; i<ROW; ++i) {
    for (j=0; j<COL; ++j) {
      printf("%2d ", j);
    }
    puts("");
  }
  printf("\n\n");

  printf("\n(i,j)\n");
  for (i=0; i<ROW; ++i) {
    for (j=0; j<COL; ++j) {
      printf("(%2d,%2d) ", i, j);
    }
    puts("");
  }
  printf("\n\n");

  printf("\nA[i][j]\n");
  for (i=0; i<ROW; ++i) {
    for (j=0; j<COL; ++j) {
      printf("%2d ", A[i][j]);
    }
    puts("");
  }
  printf("\n\n");

}
```



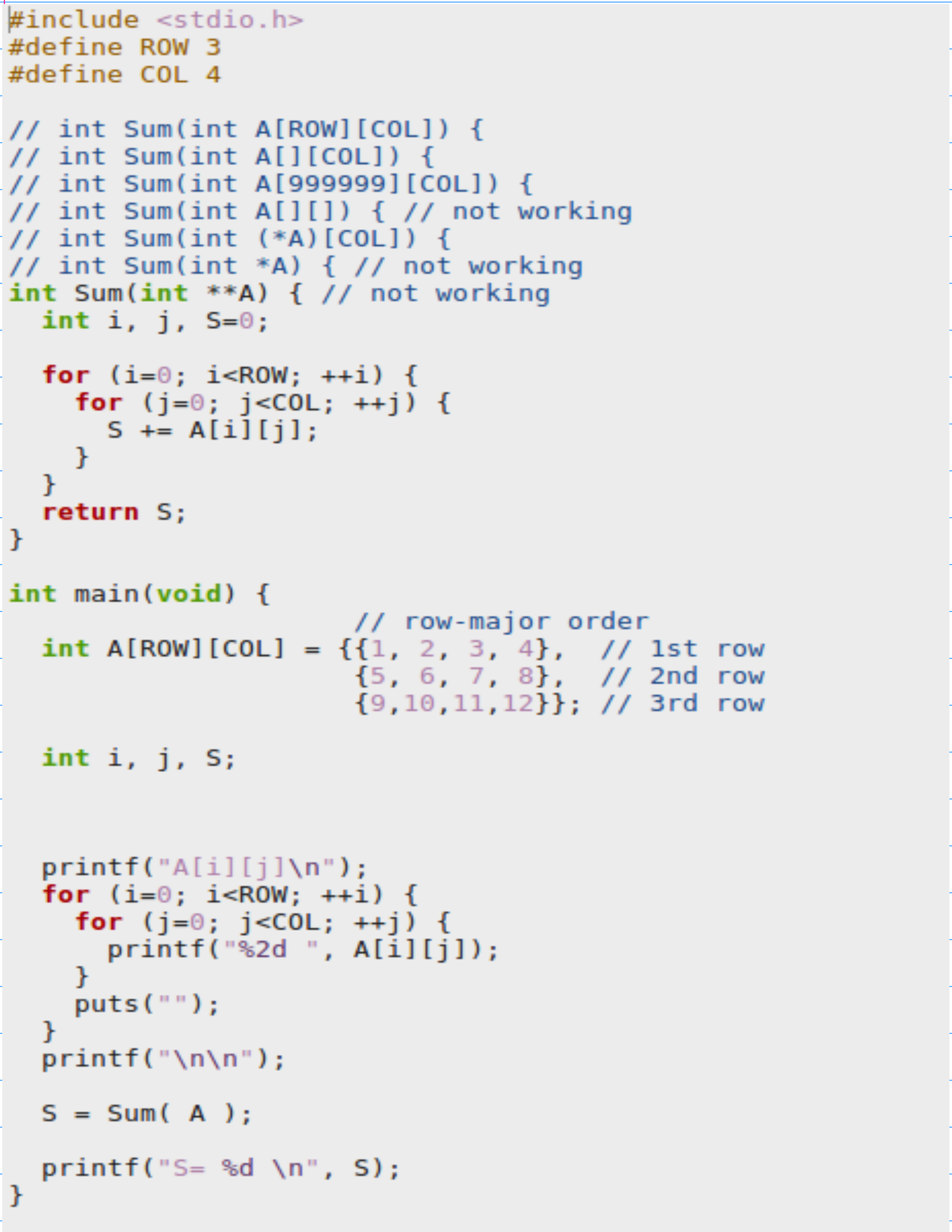
```
( 0, 0) ( 0, 1) ( 0, 2) ( 0, 3)
( 1, 0) ( 1, 1) ( 1, 2) ( 1, 3)
( 2, 0) ( 2, 1) ( 2, 2) ( 2, 3)



i
 0  0  0  0
 1  1  1  1
 2  2  2  2



j
 0  1  2  3
 0  1  2  3
 0  1  2  3



(i,j)
( 0, 0) ( 0, 1) ( 0, 2) ( 0, 3)
( 1, 0) ( 1, 1) ( 1, 2) ( 1, 3)
( 2, 0) ( 2, 1) ( 2, 2) ( 2, 3)



A[i][j]
 1  2  3  4
 5  6  7  8
 9 10 11 12
```



```c
#include <stdio.h>
#define ROW 3
#define COL 4

int main(void) {
                     // row-major order
  int A[ROW][COL] = {{1, 2, 3, 4},  // 1st row
                     {5, 6, 7, 8},  // 2nd row
                     {9,10,11,12}}; // 3rd row

  int i, j;



  printf("A[i][j]");
  for (i=0; i<ROW; ++i) {
    for (j=0; j<COL; ++j) {
      printf("%2d ", A[i][j]);
    }
    puts("");
  }
  printf("\n\n");


  printf("A[i][j]\n");
  for (i=0; i<ROW; ++i) {
    for (j=0; j<COL; ++j) {
      printf("&A[%d][%d]= %p \n", i, j, &A[i][j]);
    }
    puts("");
  }
  printf("\n\n");


  printf("A[i]\n");
  for (i=0; i<ROW; ++i) {
    printf("A[%d]= %p \n", i, A[i]);
  }
  printf("\n\n");


  printf("A= %p \n", A);

}
```



```
A[i][j] 1  2  3  4
 5  6  7  8
 9 10 11 12


A[i][j]
&A[0][0]= 0x7ffc045ced70
&A[0][1]= 0x7ffc045ced74
&A[0][2]= 0x7ffc045ced78
&A[0][3]= 0x7ffc045ced7c

&A[1][0]= 0x7ffc045ced80
&A[1][1]= 0x7ffc045ced84
&A[1][2]= 0x7ffc045ced88
&A[1][3]= 0x7ffc045ced8c

&A[2][0]= 0x7ffc045ced90
&A[2][1]= 0x7ffc045ced94
&A[2][2]= 0x7ffc045ced98
&A[2][3]= 0x7ffc045ced9c



A[i]
A[0]= 0x7ffc045ced70
A[1]= 0x7ffc045ced80
A[2]= 0x7ffc045ced90


A= 0x7ffc045ced70
```



```c
#include <stdio.h>
#define ROW 3
#define COL 4

// int Sum(int A[ROW][COL]) {
// int Sum(int A[][COL]) {
// int Sum(int A[999999][COL]) {
// int Sum(int A[][]) { // not working
// int Sum(int (*A)[COL]) {
// int Sum(int *A) { // not working
int Sum(int **A) { // not working
  int i, j, S=0;

  for (i=0; i<ROW; ++i) {
    for (j=0; j<COL; ++j) {
      S += A[i][j];
    }
  }
  return S;
}

int main(void) {
                     // row-major order
  int A[ROW][COL] = {{1, 2, 3, 4},  // 1st row
                     {5, 6, 7, 8},  // 2nd row
                     {9,10,11,12}}; // 3rd row

  int i, j, S;



  printf("A[i][j]\n");
  for (i=0; i<ROW; ++i) {
    for (j=0; j<COL; ++j) {
      printf("%2d ", A[i][j]);
    }
    puts("");
  }
  printf("\n\n");

  S = Sum( A );

  printf("S= %d \n", S);
}
```