Naseem Auguste

COP 3502 Exercise-Pointer and Array Revision

This exercise covers pointers, arrays and 2D arrays. Try to solve the following exercise without using any compiler.

Pointers

1.)What would be the output of the following programs:

(a)

```
int main( )
{
      int i = 5, j = 2 ;
      junk ( i, j ) ;
      printf ( "\n%d %d", i, j ) ;
      return 0;
}
void junk ( int i, int j )
{
      i = i * i ;
      j = j * j ;
}
```

5 2

(b)

```
 int main( )
{
      int i = 5, j = 2 ;
      junk ( &i, &j ) ;
      printf ( "\n%d %d", i, j ) ;
      return 0;
}
void junk ( int *i, int *j )
{
      *i = *i * *i ;
      *j = *j * *j ;
}
```

25 4

(c)

```
int main( )
{
      int i = 4, j = 2 ;
      junk ( &i, j ) ;
      printf ( "\n%d %d", i, j ) ;
      return 0;
}
void junk ( int *i, int j )
{
      *i = *i * *i ;
      j = j * j ;
}
```

16 2

1006   1006   1006

13 5   135   135   13 5   13.5

```
(d)
int main( )
{
      float a = 13.5 ;
      float *b, *c ;
      b = &a ; /* suppose address of a is 1006 */
      c = b ;
      printf ( "\n%u %u %u", &a, b, c ) ;
      printf ( "\n%f %f %f %f %f", a, *(&a), *&a, *b, *c ) ;
      return 0;
}
```

2.)Point out the errors, if any, in the following programs:

```
(a)
int main( )
{
      int i = 135, a = 135, k ;
      k = pass ( i, a ) ;
      printf ( "\n%d", k ) ;
      return 0;
}
int pass ( int j, int b )
int c ;
{
      c = j + b ;
      return ( c ) ;
}
```

move here

```
(b)
int main( )
{
      int p = 23, f = 24 ;
      jiaayjo ( &p, &f ) ;
      printf ( "\n%d %d", p, f ) ;
      return 0;
}
void jiaayjo ( int q, int g )
{
      q = q + q ;
      g = g + g ;
}
```

none

```
(c)
int main( )
{
      int k = 35, z ;
      z = check ( k ) ;
      printf ( "\n%d", z ) ;
      return 0;
}
int check ( m )
```

```
{
      int m ;
      if ( m > 40 )
          return ( 1 ) ;
      else
          return ( 0 ) ;
}
```

← not needed

```
(d)
int main( )
{
      int i = 35, *z ;
      z = function ( &i ) ;
      printf ( "\n%d", z ) ;
      return 0;
}
int* function ( int *m )
{
      return ( m + 2 ) ;
}
```

not needed ↓

Arrays and Pointers

3.)What would be the output of the following programs:

```
10 10    32 30    54 50
21 20    43 40
```

```
(a)
int main( )
{
      int b[ ] = { 10, 20, 30, 40, 50 } ;
      int i ;
      for ( i = 0 ; i <= 4 ; i++ )
            printf ( "\n%d %d", *( b + i ), b[i] ) ;

      return 0;
}
(b)
int main( )
{
  int b[ ] = { 0, 20, 0, 40, 5 } ;
  int i, *k, *ar ;
  k = b ;
  ar = b;
  for ( i = 0 ; i <= 4 ; i++ )
  {
    printf ( "\n%d %d", *k, ar[i] ) ;
    k++ ;
  }
  return 0;
}
```

```
0 0
20 20
0 0
40 40
5 5
```

(c)

```
int main( )
{
      int a[ ] = { 2, 4, 6, 8, 10 } ;
      int i ;
      change ( a, 5 ) ;
      for ( i = 0 ; i <= 4 ; i++ )
          printf( "\n%d", a[i] ) ;
      return 0;
}
void change ( int *b, int n )
{
      int i ;
      for ( i = 0 ; i < n ; i++ )
            *( b + i ) = *( b + i ) + 5 ;
}
```

7
9
11
13
15

(d)

```
int  main( )
{
      int a[5], i, b = 16 ;
      for ( i = 0 ; i < 5 ; i++ )
            a[i] = 2 * i ;
      f ( a, b ) ;
      for ( i = 0 ; i < 5 ; i++ )
      printf ( "\n%d", a[i] ) ;
      printf( "\n%d", b ) ;
}
void f ( int *x, int y )
{
      int i ;
      for ( i = 0 ; i < 5 ; i++ )
      *( x + i ) += 2 ;
      y += 2 ;
}
```

2
4
6
8
10
16

(e)

```
int main( )
{
      static int a[5] ;
      int i ;
      for ( i = 0 ; i <= 4 ; i++ )
      printf ( "\n%d", a[i] ) ;
      return 0;
}
```

0 0 0 0 0

```
(f)
int main( )
{
      int a[5] = { 5, 1, 15, 20, 25 } ;
      int i, j, k = 1, m ;
      i = ++a[1] ;
      j = a[1]++ ;
      m = a[i++] ;
      printf ( "\n%d %d %d", i, j, m ) ;
      return 0;
}
```

3 2 15

4.What would be the output of the following programs:

```
(a)
int main( )
{
      int n[3][3] = {
                        2, 4, 3,
                        6, 8, 5,
                        3, 5, 1 } ;
      printf ( "\n%d %d %d", *n, n[3][3], n[2][2] ) ;
      return 0;
}
```

Garbage Value

Null Char

1

```
(b)
main( )
{
      int n[3][3] = {
                        2, 4, 3,
                        6, 8, 5,
                        3, 5, 1  } ;
      int i, *ptr ;

      ptr = n ;
      for ( i = 0 ; i <= 8 ; i++ )
            printf ( "\n%d", *( ptr + i ) ) ;
}
```

2 4 3 6 8 5 3

5 1

```
(c)
int main( )
{
      int n[3][3] = {
                  2, 4, 3,
                  6, 8, 5,
                  3, 5, 1 } ;
      int i, j ;
      for ( i = 0 ; i <= 2 ; i++ )
            for ( j = 0 ; j <= 2 ; j++ )
                  printf ( "\n%d %d", n[i][j], *( *( n + i ) + j ) ) ;
      return 0
}
```

2 2
4 4
3 3
6 6
8 8

5 5
3 3
5 5
1 1

Same

5.) **Match the following with reference (1 to 10) to the following segment (a to m):**

```
int x[3][5] = {    0  1  2  3  4
                 0{ 1, 2, 3, 4, 5 },
                 1{ 6, 7, 8, 9, 10 },
                 2{ 11, 12, 13, 14, 15 }
               };
int *n = x ;
```

1	*(*(x + 2) + 1) f	a.	9
2	J *(*x + 2) + 5	b.	13
3	*(*(x + 1)) m	c.	4
4	c *(*(x) + 2) + 1	d.	3
5	* (*(x + 1) + 3) a	e.	2
6	I *n	f.	12
7	*(n +2) D	g.	14
8	k (*(n + 3) + 1	h.	7
9	*(n + 5)+1 h	i.	1
10	e ++*n	j.	8
		k.	5
		l.	10
		m.	6

6. If you want to store an array of 5 strings with the maximum length of a string is 100, how would you declare it?

Answer: char howdy [5][100]

7. Given a 2D array. Write a function that will take the 2D array as argument and then print the maximum value in each row of that 2D array. Also, write a line to show how would you call that function. Do not write the main function.

```
#define ROW 5
#define COL 6
void printMax(int values [][6], int len),
void printMax(int values [][6], int len)
{
  int i, j, max =0,
  for(i=0, i<ROW, i++)     { if(values [i][j]>max
     for(j=0, j<COL, j++)  {  max =values[i][j],}
  printf("%d", max),}
```