Arrays (1B)

a1.c

```
#include <stdio.h>

void main(void) {
  int a[5] = {10, 20, 30, 40, 50};
  int *p = a;
  int i;

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

  for (i=0; i<5; ++i) {
    printf("&a[%d]= %p  ",  i , &a[i]);
    printf("a[%d]= %d  ",   i ,  a[i]);
    printf("(a+%d)= %p  ",  i ,  (a+i));
    printf("*(a+%d)= %d\n", i , *(a+i));
  }
  printf("\n");

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

  for (i=0; i<5; ++i) {
    printf("&p[%d]= %p  ",  i , &p[i]);
    printf("p[%d]= %d  ",   i ,  p[i]);
    printf("(p+%d)= %p  ",  i ,  (p+i));
    printf("*(p+%d)= %d\n", i , *(p+i));
  }
  printf("\n");


}
```

```
a= 0xbfa91298
&a[0]= 0xbfa91298  a[0]= 10  (a+0)= 0xbfa91298  *(a+0)= 10
&a[1]= 0xbfa9129c  a[1]= 20  (a+1)= 0xbfa9129c  *(a+1)= 20
&a[2]= 0xbfa912a0  a[2]= 30  (a+2)= 0xbfa912a0  *(a+2)= 30
&a[3]= 0xbfa912a4  a[3]= 40  (a+3)= 0xbfa912a4  *(a+3)= 40
&a[4]= 0xbfa912a8  a[4]= 50  (a+4)= 0xbfa912a8  *(a+4)= 50

&p= 0xbfa91294 p= 0xbfa91298
&p[0]= 0xbfa91298  p[0]= 10  (p+0)= 0xbfa91298  *(p+0)= 10
&p[1]= 0xbfa9129c  p[1]= 20  (p+1)= 0xbfa9129c  *(p+1)= 20
&p[2]= 0xbfa912a0  p[2]= 30  (p+2)= 0xbfa912a0  *(p+2)= 30
&p[3]= 0xbfa912a4  p[3]= 40  (p+3)= 0xbfa912a4  *(p+3)= 40
&p[4]= 0xbfa912a8  p[4]= 50  (p+4)= 0xbfa912a8  *(p+4)= 50
```

a2.c

```c
#include <stdio.h>

void main (void) {
  int  c[3][4] = {{1, 2, 3, 4},
                  {5, 6, 7, 8},
                  {9,10,11,12}};
  int i, j;

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

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

  for (i=0; i<3; i++) {
      printf("&c[%d]= %p  ", i, &c[i]);
      printf("c[%d]= %p\n ", i, c[i]);
    printf("-----------------------------\n");
  }
  printf("\n");




}
```

```
  1   2   3   4
  5   6   7   8
  9  10  11  12

&c[0][0]= 0xbfb315b8  c[0][0]= 1
&c[0][1]= 0xbfb315bc  c[0][1]= 2
&c[0][2]= 0xbfb315c0  c[0][2]= 3
&c[0][3]= 0xbfb315c4  c[0][3]= 4
-----------------------------
&c[1][0]= 0xbfb315c8  c[1][0]= 5
&c[1][1]= 0xbfb315cc  c[1][1]= 6
&c[1][2]= 0xbfb315d0  c[1][2]= 7
&c[1][3]= 0xbfb315d4  c[1][3]= 8
-----------------------------
&c[2][0]= 0xbfb315d8  c[2][0]= 9
&c[2][1]= 0xbfb315dc  c[2][1]= 10
&c[2][2]= 0xbfb315e0  c[2][2]= 11
&c[2][3]= 0xbfb315e4  c[2][3]= 12
-----------------------------

&c[0]= 0xbfb315b8  c[0]= 0xbfb315b8
 -----------------------------
&c[1]= 0xbfb315c8  c[1]= 0xbfb315c8
 -----------------------------
&c[2]= 0xbfb315d8  c[2]= 0xbfb315d8
 -----------------------------
```

a3.c

```
      printf("(*(c+%d)+%d)= %p  ", i, j, (*(c+i)+j));
      printf("*(*(c+%d)+%d)= %d\n", i, j, *(*(c+i)+j));

      printf("(c+%d)= %p  ", i, (c+i));
      printf("*(c+%d= %p\n ", i, *(c+i));
```

```
&c[0]= 0xbfaa03a8  c[0]= 0xbfaa03a8
 -----------------------------
&c[1]= 0xbfaa03b8  c[1]= 0xbfaa03b8
 -----------------------------
&c[2]= 0xbfaa03c8  c[2]= 0xbfaa03c8
 -----------------------------

(*(c+0)+0)= 0xbfaa03a8  *(*(c+0)+0)= 1
(*(c+0)+1)= 0xbfaa03ac  *(*(c+0)+1)= 2
(*(c+0)+2)= 0xbfaa03b0  *(*(c+0)+2)= 3
(*(c+0)+3)= 0xbfaa03b4  *(*(c+0)+3)= 4
-----------------------------
(*(c+1)+0)= 0xbfaa03b8  *(*(c+1)+0)= 5
(*(c+1)+1)= 0xbfaa03bc  *(*(c+1)+1)= 6
(*(c+1)+2)= 0xbfaa03c0  *(*(c+1)+2)= 7
(*(c+1)+3)= 0xbfaa03c4  *(*(c+1)+3)= 8
-----------------------------
(*(c+2)+0)= 0xbfaa03c8  *(*(c+2)+0)= 9
(*(c+2)+1)= 0xbfaa03cc  *(*(c+2)+1)= 10
(*(c+2)+2)= 0xbfaa03d0  *(*(c+2)+2)= 11
(*(c+2)+3)= 0xbfaa03d4  *(*(c+2)+3)= 12
-----------------------------

(c+0)= 0xbfaa03a8  *(c+0= 0xbfaa03a8
 -----------------------------
(c+1)= 0xbfaa03b8  *(c+1= 0xbfaa03b8
 -----------------------------
(c+2)= 0xbfaa03c8  *(c+2= 0xbfaa03c8
 -----------------------------
```

a4.c

```
  int *p = c[0];


      printf("(p+%d*4+%d)= %p  ", i, j, (p+i*4+j));
      printf("*(p+%d*4+%d)= %d\n", i, j, *(p+i*4+j));

      printf("&p[%d*4+%d]= %p  ", i, j, &p[i*4+j]);
      printf("p[%d*4+%d]= %d\n", i, j, p[i*4+j]);
```

```
&c[0]= 0xbfe2df04  c[0]= 0xbfe2df04
 -----------------------------
&c[1]= 0xbfe2df14  c[1]= 0xbfe2df14
 -----------------------------
&c[2]= 0xbfe2df24  c[2]= 0xbfe2df24
 -----------------------------

(p+0*4+0)= 0xbfe2df04  *(p+0*4+0)= 1
(p+0*4+1)= 0xbfe2df08  *(p+0*4+1)= 2
(p+0*4+2)= 0xbfe2df0c  *(p+0*4+2)= 3
(p+0*4+3)= 0xbfe2df10  *(p+0*4+3)= 4
-----------------------------
(p+1*4+0)= 0xbfe2df14  *(p+1*4+0)= 5
(p+1*4+1)= 0xbfe2df18  *(p+1*4+1)= 6
(p+1*4+2)= 0xbfe2df1c  *(p+1*4+2)= 7
(p+1*4+3)= 0xbfe2df20  *(p+1*4+3)= 8
-----------------------------
(p+2*4+0)= 0xbfe2df24  *(p+2*4+0)= 9
(p+2*4+1)= 0xbfe2df28  *(p+2*4+1)= 10
(p+2*4+2)= 0xbfe2df2c  *(p+2*4+2)= 11
(p+2*4+3)= 0xbfe2df30  *(p+2*4+3)= 12
-----------------------------

&p[0*4+4]= 0xbfe2df14  p[0*4+4]= 5
-----------------------------
&p[1*4+4]= 0xbfe2df24  p[1*4+4]= 9
-----------------------------
&p[2*4+4]= 0xbfe2df34  p[2*4+4]= 2
-----------------------------
```

a5.c

```c
#include <stdio.h>

int main(void)
{

  int i;
  int x[5];  // x[0], x[1], x[2], x[3], x[4]

  x[0] = 10;
  x[1] = 20;
  x[2] = 30;
  x[3] = 40;
  x[4] = 50;

  printf("x[0]= %d \n", x[0]);
  printf("x[1]= %d \n", x[1]);
  printf("x[2]= %d \n", x[2]);
  printf("x[3]= %d \n", x[3]);
  printf("x[4]= %d \n", x[4]);

  x[0] += 1;
  x[1] += 2;
  x[2] += 3;
  x[3] += 4;
  x[4] += 5;

  for (i=0; i<5; ++i) {
    printf("x[%d]= %d \n", i, x[i]);
  }


  printf("addr(i)=%p data(i)=%d \n", &i, i);

  for (i=0; i<5; ++i) {
    printf("addr(x[%d])= %p \n", i, &(x[i]));
  }

  printf("sizeof(int) = %ld bytes \n", sizeof(int));
  printf("sizeof(short) = %ld bytes \n", sizeof(short));
  printf("sizeof(long int) = %ld bytes \n", sizeof(long int));

}
```