数组、指针笔试题

```c
//一维数组
int a[] = {1,2,3,4};
printf("%d\n",sizeof(a));
printf("%d\n",sizeof(a+0));
printf("%d\n",sizeof(*a));
printf("%d\n",sizeof(a+1));
printf("%d\n",sizeof(a[1]));
printf("%d\n",sizeof(&a));
printf("%d\n",sizeof(*&a));
printf("%d\n",sizeof(&a+1));
printf("%d\n",sizeof(&a[0]));
printf("%d\n",sizeof(&a[0]+1));

//字符数组
char arr[] = {'a','b','c','d','e','f'};
printf("%d\n", sizeof(arr));
printf("%d\n", sizeof(arr+0));
printf("%d\n", sizeof(*arr));
printf("%d\n", sizeof(arr[1]));
printf("%d\n", sizeof(&arr));
printf("%d\n", sizeof(&arr+1));
printf("%d\n", sizeof(&arr[0]+1));

printf("%d\n", strlen(arr));
printf("%d\n", strlen(arr+0));
printf("%d\n", strlen(*arr));
printf("%d\n", strlen(arr[1]));
printf("%d\n", strlen(&arr));
printf("%d\n", strlen(&arr+1));
printf("%d\n", strlen(&arr[0]+1));

char arr[] = "abcdef";
printf("%d\n", sizeof(arr));
printf("%d\n", sizeof(arr+0));
printf("%d\n", sizeof(*arr));
printf("%d\n", sizeof(arr[1]));
printf("%d\n", sizeof(&arr));
printf("%d\n", sizeof(&arr+1));
printf("%d\n", sizeof(&arr[0]+1));

printf("%d\n", strlen(arr));
printf("%d\n", strlen(arr+0));
printf("%d\n", strlen(*arr));
printf("%d\n", strlen(arr[1]));
printf("%d\n", strlen(&arr));
printf("%d\n", strlen(&arr+1));
printf("%d\n", strlen(&arr[0]+1));

char *p = "abcdef";
printf("%d\n", sizeof(p));
printf("%d\n", sizeof(p+1));
printf("%d\n", sizeof(*p));
printf("%d\n", sizeof(p[0]));
printf("%d\n", sizeof(&p));
printf("%d\n", sizeof(&p+1));
printf("%d\n", sizeof(&p[0]+1));

printf("%d\n", strlen(p));
printf("%d\n", strlen(p+1));
printf("%d\n", strlen(*p));
printf("%d\n", strlen(p[0]));
printf("%d\n", strlen(&p));
printf("%d\n", strlen(&p+1));
printf("%d\n", strlen(&p[0]+1));

//二维数组
int a[3][4] = {0};
printf("%d\n",sizeof(a));
printf("%d\n",sizeof(a[0][0]));
printf("%d\n",sizeof(a[0]));
printf("%d\n",sizeof(a[0]+1));
printf("%d\n",sizeof(*(a[0]+1)));
printf("%d\n",sizeof(a+1));
printf("%d\n",sizeof(*(a+1)));
printf("%d\n",sizeof(&a[0]+1));
printf("%d\n",sizeof(*(&a[0]+1)));
printf("%d\n",sizeof(*a));
printf("%d\n",sizeof(a[3]));
```

```
int main()
{
    int a[5] = { 1, 2, 3, 4, 5 };
    int *ptr = (int *)(&a + 1);
    printf( "%d,%d", *(a + 1), *(ptr - 1));
    return 0;
}
//程序的结果是什么？
```

```
struct Test
{
    int Num;
    char *pcName;
    short sDate;
    char cha[2];
    short sBa[4];
}*p;
//假设p 的值为0x100000。 如下表表达式的值分别为多少？
int main()
{
    printf("%p\n", p + 0x1);
    printf("%p\n", (unsigned long)p + 0x1);
    printf("%p\n", (unsigned int*)p + 0x1);
    return 0;
}
```

```
int main()
{
    int a[4] = { 1, 2, 3, 4 };
    int *ptr1 = (int *)(&a + 1);
    int *ptr2 = (int *)((int)a + 1);
    printf( "%x,%x", ptr1[-1], *ptr2);
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    int a[3][2] = { (0, 1), (2, 3), (4, 5) };
    int *p;
    p = a[0];
    printf( "%d", p[0]);
    return 0;
}
```

```
int main()
{
    int a[5][5];
    int(*p)[4];
    p = a;
    printf( "%p,%d\n", &p[4][2] - &a[4][2], &p[4][2] - &a[4][2]);
    return 0;
}
```

```
int main()
{
    int aa[2][5] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    int *ptr1 = (int *)(&aa + 1);
    int *ptr2 = (int *)(*(aa + 1));
    printf( "%d,%d", *(ptr1 - 1), *(ptr2 - 1));
    return 0;
}
```

```
#include <stdio.h>

int main()
{
    char *a[] = {"work","at","alibaba"};
    char**pa = a;
    pa++;
    printf("%s\n", *pa);
    return 0;
}
```

```
int main()
{
    char *c[] = {"ENTER","NEW","POINT","FIRST"};
    char**cp[] = {c+3,c+2,c+1,c};
    char***cpp = cp;
    printf("%s\n", **++cpp);
    printf("%s\n", *--*++cpp+3);
    printf("%s\n", *cpp[-2]+3);
    printf("%s\n", cpp[-1][-1]+1);
    return 0;
}
```