A3-32

```
unsigned char array_1[1024] = {1};
unsigned char array_2[1024] = {1};
unsigned char array_3[1024] = {1};
unsigned char array_4[1024] = {1};
unsigned char array_5[1024];
unsigned char array_6[1024];
unsigned char array_7[1024];
unsigned char array_8[1024];

int main()
{
        return 0;
}
```

A3-33

```
char array[1024UL*1024UL];

int main()
{
        return 0;
}
```

A3-36

```
char some_var[1024ULL*1024ULL*1024ULL * 1ULL] = {0x41};
int main()
{
        return 0;
}
```

A3-37

```
int main()
{
        printf("%p",malloc(4UL*1024UL*1024UL*1024UL));
        return 0;
}
```

A3-39

```
int main()
{
        printf("%p\n", malloc(2U*1024U));
        return 0;
}
```

A3-44

```
static int stat = 8;
int global = 8;
int num() {return 0;}
int main()
{
        printf("function %p\n", &num);
        printf("global %p\n", &global);
        printf("static %p\n", &stat);
        return 0;
}
```

A3-45

```
int uninitialized;
int initialized = 1;
int num() { return 1;}
```

```
int main()
{
        printf("Initialized: %p\n", &initialized);
        printf("Uninitialized: %p\n", &uninitialized);
        printf("function :%p\n",&num);

        return 0;
}
```

A3-46

```
int main()
{
        malloc(8UL*1024UL*1024UL*1024UL);

        return 0;
}
```

A3-49

```
int main()
{
        malloc(1024*1024);

        return 0;
}
```

A3-53

```
int main()
{
        char* i = 0;
        printf("%c", i[20]);
        return 0;
}
```

A3-55

```
int main()
{
        printf("%p", &getchar);
        return 0;
}
```

A3-57

```
int main()
{
        int local = 8;
        printf("%p", &local);
        return 0;
}
```

A3-58

```
int global = 8;
int main()
{
        printf("%p", &global);
        return 0;
}
```

A3-59

```
int main()
{
        int *n = malloc(10);
        printf("%p", n);
        return 0;
}
```

A3-70

```
char global[256UL*1024UL*1024UL];
int main()
{
        int* maloc = malloc(256UL*1024UL*1024UL);
        extern int exte;
        return 0;
}
```

A3-78

```
int global = 8;
int main()
{
        int* heap = malloc(1 * sizeof(int));
        *heap = 7;
        int local = 23;
        int space = shm_open("name", O_CREAT|O_RDWR, 0666 );
        ftruncate(space, sizeof(int) * 20);
        int* shared = mmap(0 , sizeof(int) * 20, PROT_READ|PROT_WRITE, MAP_SHARED, space, 0);
        int sum = local + global + *heap ;
        printf("%d", sum);

        return 0;
}
```

A3-86

```
int main()
{
        const int num = 3;
        int* p = &num;
        *p = 8;
        printf("%d",num);
        return 0;
}
```