

3.71/

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define BufferSize 8

int good_echo()
{
    char* buffer = (char*)calloc(BufferSize,sizeof(char));
    if (buffer == NULL)
    {
        fprintf(stderr, "Error: failed to allocate buffer.\n");
        return -1;
    }
    while (1)
    {
        fgets(buffer, BufferSize, stdin);
        if (strlen(buffer) == BufferSize - 1)
        {
            fputs(buffer, stdout);
            if (buffer[BufferSize - 1 - 1] == '\n')
            {
                break;
            }
            memset(buffer, 0, strlen(buffer));
        }
        else if (strlen(buffer) < BufferSize - 1)
        {
            fputs(buffer, stdout);
            break;
        }
    }
    free(buffer);
    return 0;
}

int main()
{
    good_echo();
}
```

3.72/

A.

$s2 = s1 - [(8 * n + 30) \& (-16)]$

作用： $8n+30$ 对 16 取整
n 为奇数时： $s2=s1-(8*n+24)$
n 为偶数时： $s2=s1-(8*n+16)$

B.
 $p=(s2+15)\&(-16)$
作用： 对 16 取整

C.
 $min=1$
 $max=24$

D.
对其到 16 的整数倍

3.75/

n	Real Part	Imaginary Part
1	%xmm0	%xmm1
2	%xmm2	%xmm3
3	%xmm4	%xmm5
n

返回%xmm0 代表实部， %xmm1 代表虚部