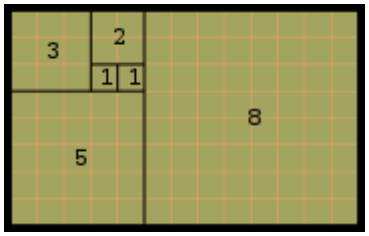
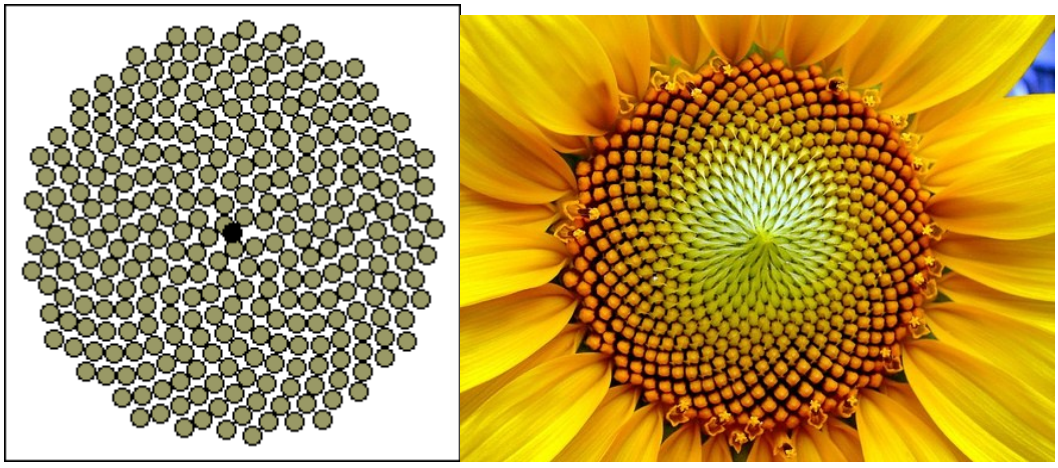
```
/* Program to count the no of positive and negative numbers*/
#include<stdio.h>

void main( )
{

      int a[50],n,count_neg=0,count_pos=0,I;
      printf("Enter the size of the array\n");
      scanf("%d",&n);
      printf("Enter the elements of the array\n");
      for I=0;I < n;I++)
            scanf("%d",&a[I]);
      for(I=0;I < n;I++)
      {

            if(a[I] < 0)
                  count_neg++;
            else
                  count_pos++;
      }
      printf("There are %d negative numbers in the
            array\n",count_neg);
      printf("There are %d positive numbers in the
            array\n",count_pos);
}
```

```
#include <stdio.h>

void print_converted(int pounds)
/* Convert U.S. Weight to Imperial and International
   Units. Print the results */
{       int stones = pounds / 14;
        int uklbs = pounds % 14;
        float kilos_per_pound = 0.45359;
        float kilos = pounds * kilos_per_pound;

        printf("   %3d          %2d  %2d        %6.2f\n",
                pounds, stones, uklbs, kilos);
}

main()
{       int us_pounds;

        printf(" US lbs      UK st. lbs       INT Kg\n");

        for(us_pounds=10; us_pounds < 250; us_pounds+=10)
                print_converted(us_pounds);
}
```

$$F_n = \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F_{n-1} + F_{n-2} & \text{if } n > 1. \end{cases}$$

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765





Ian Stewart, "Life's other secret", Wiley 1999

```
#include <stdio.h>

main()
{       int fib[24];
        int i;

        fib[0] = 0;
        fib[1] = 1;

        for(i = 2; i < 24; i++)
                fib[i] = fib[i-1] + fib[i-2];

        for (i = 0; i < 24; i++)
                printf("%3d   %6d\n", i, fib[i]);
}
```

BUILT IN FUNCTIONS FOR CHARACTER HANDLING

The following character handling functions are defined in ctype.h

isalnum	*Tests for alphanumeric character*
isalpha	*Tests for alphabetic character*
isascii	*Tests for ASCII character*
iscntrl	*Tests for control character*
isdigit	*Tests for 0 to 9*
isgraph	*Tests for printable character*
islower	*Tests for lowercase*
isprint	*Tests for printable character*
ispunct	*Tests for punctuation character*

isspace	*Tests for space character*
isupper	*Tests for uppercase character*
isxdigit	*Tests for hexadecimal*
toascii	*Converts character to ascii code*
tolower	*Converts character to lowercase*
toupper	*Converts character to uppercase*

```
/* addn.c -- Read a positive number N. Then read N integers and
 *           print them out together with their sum.
 */

#include <stdio.h>

int main(void) {
  int n;       /* The number of numbers to be read */
  int sum;     /* The sum of numbers already read  */
  int current; /* The number just read             */
  int lcv;     /* Loop control variable, it counts the number
                  of numbers already read */

  printf("Enter a positive number n > ");
  scanf("%d",&n); /* We should check that n is really positive*/
  sum = 0;
  for (lcv=0; lcv < n; lcv++) {
    printf("\nEnter an integer > ");
    scanf("%d",&current);
    /*    printf("\nThe number was %d\n", current); */
    sum = sum + current;
  }
  printf("The sum is %d\n", sum);
  return 0;
}
```

```
/* prime1.c  It prompts the user to enter an integer N. It prints out
 *           if it is a prime or not. If not, it prints out a factor
of N.
 */

#include <stdio.h>

int main(void) {
  int n;
  int i;
  int flag;

  printf("Enter value of N > ");
  scanf("%d", &n);
  flag = 1;
  for (i=2; (i<(n/2)) && flag; ) { /* May be we do not need to test
              values of i greater than the square root of n? */
    if ((n % i) == 0) /* If true n is divisible by i */
      flag = 0;
    else
      i++;
  }

  if (flag)
    printf("%d is prime\n", n);
```

```
  else
    printf("%d has %d as a factor\n", n, i);
  return 0;
}


/* line.c -- It reads lines from input and echoes them back.
 */

#include <stdio.h>

int main(void) {
  char c;
  int count;

  for(;;){
    count=0;
    printf("Please enter a line [blank line to terminate]> ");
    do{
      c=getchar();
      putchar(c);
      count++;
    }while (c!='\n');
    if(count==1)break;
  }
}
```