

Указатели на функции. Пример использования

З а д а ч а. Напечатать таблицу значений функций \sin , \cos , \log в целых точках отрезка $[1, N]$ с шагом 1. Значения печатать с двумя знаками после запятой.

Решение

Воспользуемся массивом структур с двумя членами: один будет содержать указатель на имя функции, другой — указатель на саму функцию. Внешний цикл будет проходить по элементам массива и печатать имя текущей функции, а внутренний цикл будет проходить по точкам отрезка и для каждой из них вызывать текущую функцию через указатель. Кроме того, необходимо сформировать заголовок таблицы. Каждое значение будем выводить в поле из 7 позиций, включающее пробел и знак вертикальной черты справа. С учетом имен функций ширина заголовка и строк таблицы составит $7*(N+1)$ позиций. Данное решение легко модифицировать, если понадобится обрабатывать еще какие-нибудь стандартные математические функции — достаточно добавить в массив структур соответствующие элементы. Других изменений в программу вносить не придется.

В конец массива добавим структуру с двумя пустыми указателями — такая организация позволяет использовать *while*-цикл для прохода по массиву, т.е. переходить к следующему элементу на очередной итерации, пока не достигнем структуры с нулевыми указателями. Однако в данном решении мы используем *for*-цикл, вычислив заранее количество функций, хранящихся в массиве, с помощью константного выражения с операцией *sizeof*.

```

/* вычислить значения функций в
   точках beg, beg+step, ..., beg+step*(N-1) */

#include <stdio.h>
#include <math.h> /* стандартный заголовок матем. функций */
#define N 5      /* количество целых точек на отрезке, N<=10 */

struct Func { char* name; /* имя функции */
              double (*fp)(double); /* адрес функции */
            }
            func [] = {
                { "sin", &sin },
                { "cos", &cos },
                { "log", &log },
                { NULL, NULL }
            };

int main()
{
    int beg = 1, /* начальная точка - левый конец отрезка */
        step = 1, /* расстояние между соседними точками */
        end = beg + (N-1)*step, /* конечная точка - правый конец отрезка */
        numfuncs = sizeof(func)/sizeof(struct Func) - 1; /* количество функций в массиве func */
    int i; int j;

```

```

/*----- печать заголовка таблицы значений -----*/
printf("\n");
for (i=1; i<=N+1;i++)
    printf("%7s", "_____");

printf("\n%7s", " f(x) |");
for (i=beg; i<=end; i+=step)
    printf("%5.2f |", (double) i);

printf("\n");
for (i=1; i<=N+1;i++)
    printf("%7s", "-----");

/*----- печать значений функций -----*/
for (i=0; i < numfuncs; i++) {
    printf("\n %-5s|", func[i].name);
    for (j=beg; j<=end; j+=step)
        printf("%5.2f |", (*func[i].fp) ( (double)j) );

}
/*----- завершение печати таблицы -----*/
printf("\n");
for (i=1; i<=N+1;i++)
    printf("%7s", "-----");
printf("\n");
}

```

Результат:

f(x)	1.00	2.00	3.00	4.00	5.00
sin	0.84	0.91	0.14	-0.76	-0.96
cos	0.54	-0.42	-0.99	-0.65	0.28
log	0.00	0.69	1.10	1.39	1.61