

```
1      PROGRAM MAIN
2          PARAMETER (MAXSIZ=99)
3          REAL A(MAXSIZ)
4      10    READ (5,100,END=999) K
5      100   FORMAT(I5)
6           IF (K.LE.0.OR. K.GT.MAXSIZ) STOP
7           READ *, (A(I),I=1,K)
8           PRINT *, (A(I),I=1,K)
9           PRINT *, 'SUM=',SUM(A,K)
10          GO TO 10
11      999  PRINT *, "All Done"
12          STOP
13          END
14  C SUMMATION SUBPROGRAM
15      FUNCTION SUM(V,N)
16          REAL :: V(N) ! New style declaration
17          SUM = 0.0
18          DO 20 I = 1,N
19              SUM = SUM + V(I)
20      20    CONTINUE
21          RETURN
22          END
```

Figure A.8. FORTRAN example to sum an array.

```

Line
1 IDENTIFICATION DIVISION.
2 PROGRAM-ID. SUM-OF-PRICES.
3 AUTHOR. T-PRATT.
4 ENVIRONMENT DIVISION.
5 CONFIGURATION SECTION.
6 SOURCE-COMPUTER. CDC-CYBER.
7 OBJECT-COMPUTER. CDC-CYBER.
8 INPUT-OUTPUT SECTION.
9 FILE-CONTROL.
10     SELECT INP-DATA ASSIGN TO INPUT.
11     SELECT RESULT-FILE ASSIGN TO OUTPUT.
12 DATA DIVISION.
13 FILE SECTION.
14 FD INP-DATA LABEL RECORD IS OMITTED.
15 01 ITEM-PRICE.
16     02 ITEM PICTURE X(30).
17     02 PRICE PICTURE 9999V99.
18     02 FILLER PICTURE X(44).
19 FD RESULT-FILE LABEL RECORD IS OMITTED.
20 01 RESULT-LINE PICTURE X(132).
21 WORKING-STORAGE SECTION.
22 77 TOT PICTURE 9999999V99, VALUE 0, USAGE IS COMPUTATIONAL.
23 77 COUNT PICTURE 9999, VALUE 0, USAGE IS COMPUTATIONAL.
24 01 SUM-LINE.
25     02 FILLER VALUE ' SUM = ' PICTURE X(12).
26     02 SUM-OUT PICTURE $$,$$$,$$9.99.
27     02 FILLER VALUE ' NO. OF ITEMS = ' PICTURE X(21).
28     02 COUNT-OUT PICTURE ZZZ9.
29 01 ITEM-LINE.
30     02 ITEM-OUT PICTURE X(30).
31     02 PRICE-OUT PICTURE ZZZ9.99.
32 PROCEDURE DIVISION.
33 START.
34     OPEN INPUT INP-DATA AND OUTPUT RESULT-FILE.
35 READ-DATA.
36     READ INP-DATA AT END GO TO PRINT-LINE.
37     ADD PRICE TO TOT.
38     ADD 1 TO COUNT.
39     MOVE PRICE TO PRICE-OUT.
40     MOVE ITEM TO ITEM-OUT.
41     WRITE RESULT-LINE FROM ITEM-LINE.
42     GO TO READ-DATA.
43 PRINT-LINE.
44     MOVE TOT TO SUM-OUT.
45     MOVE COUNT TO COUNT-OUT.
46     WRITE RESULT-LINE FROM SUM-LINE.
47     CLOSE INP-DATA AND RESULT-FILE.
48 STOP RUN.

```

Fig. 13-1 Example COBOL program

Line

```
1      TEST: PROCEDURE OPTIONS (MAIN);
2          START: GET LIST (K);
3              IF K > 0 THEN BEGIN;
4                  DECLARE A(K) FLOAT;
5                  GET LIST (A);
6                  PUT LIST ('INPUT IS',A,'SUM IS',SUM(A));
7                  GO TO START;
8              END;
9          SUM: PROCEDURE (V) RETURNS (FLOAT);
10             DECLARE V(*) FLOAT,
11                 TEMP FLOAT INITIAL (0);
12             DO I = 1 TO DIM(V,1);
13                 TEMP = TEMP + V(I); END;
14             RETURN(TEMP);
15             END SUM;
16     END TEST;
```

Fig. 14-1 Example PL/I program

```
1      program main(input,output,infile);
2      const size = 99;
3      type Vector = array [1..size] of real;
4      var infile: text;
5          a: Vector;
6          j,k: integer;
7      function sum(v: Vector; n: integer): real;
8          var temp: real;
9          i: integer;
10         {Body of function sum}
11         begin
12             temp := 0;
13             for i := 1 to n do temp := temp + v[i];
14             sum := temp
15         end ;{sum}
16     begin {of main}
17         reset(infile,'sample.data');
18         while not (eof(infile)) do
19             begin
20                 read(infile, k);
21                 for j := 1 to k do
22                     begin
23                         read(infile, a[j]);
24                         write(a[j]:10:2)
25                     end;
26                 writeln;
27                 writeln('sum = ',sum(a,k):6:4);
28                 readln(infile)
29             end
30     end.
```

Figure A.13. Pascal example to sum an array.

```
1  #include <stdio.h>
2  const int maxsize=9;
3  main()
4      {int a[maxsize];
5        int j,k;
6        while( (k=convert(getchar())) != 0) {
7            for(j=0; j<k; j++) a[j] = convert(getchar());
8            for(j=0; j<k; j++) printf("%d ", a[j]);
9            printf("; SUM= %d\n", addition(a,k));
10           while(getchar() != '\n');
11           } }
12  /* Function convert subprogram */
13  int convert(char ch)
14      {return ch-'0';}
15  /* Function addition subprogram */
16  int addition(v, n)
17      int v[ ], n;
18      {int sum,j;
19        sum=0;
20        for(j=0; j<n; j++) sum=sum+v[j];
21        return sum;
22      }
```

Figure A.5. C example to sum an array.

```
1  #include <stream.h>
2      // This is C++ IO streams. stdio.h also works
3  class DataConvert {
4  protected:
5      int convert(char ch) {return ch-'0';}};

6  class DataStore: DataConvert{
7  public:
8      int initial(char a)
9          {ci=0;
10             return size = convert(a);};
11     void save(char a)
12         {store[ci++]=convert(a);};
13     int setprint() { ci=0; return size;};
14     int printval() { return store[ci++];};
15     int sum()
16         {int arrsum;
17             arrsum=0;
18             for(ci=0;ci<size;ci++)arrsum=arrsum+store[ci];
19             return arrsum;}
20 private:
21     const int maxsize=9;
22     int size; // Size of array
23     int ci; // Current index into array
24     int store[maxsize];};

25 main()
26     {int j,k;
27     DataStore x;
28     while((k=x.initial(cin.get()))!=0)
29         {for(j=0;j<k;j++)x.save(cin.get());
30         for(j=x.setprint(); j>0; j--)cout << x.printval();
31         cout << "; SUM=" << x.sum() << endl;
32         while(cin.get()!='\n');}}
```

Figure A.7. C++ example to sum an array.

```
1 package ArrayCalc is
2     type Mydata is private;
3     function sum return integer;
4     procedure setval(arg:in integer);
5     private
6         size: constant:= 99;
7         type myarray is array(1..size) of integer;
8         type Mydata is record
9             val: myarray;
10            sz: integer := 0;
11            end record;
12            v: Mydata;
13 end;
14 package body ArrayCalc is
15     function sum return integer is
16         temp: integer;
17         -- Body of function sum
18         begin
19             temp := 0;
20             for i in 1..v.sz loop
21                 temp := temp + v.val(i);
22             end loop;
23             v.sz:=0;
24             return temp;
25         end sum;
26     procedure setval(arg:in integer) is
27         begin
28             v.sz:= v.sz+1;
29             v.val(v.sz):=arg;
30         end setval; end;
31 with Text_IO; use Text_IO;
32 with ArrayCalc; use ArrayCalc;
33 procedure main is
34     k, m: integer;
35 begin -- of main
36     get(k);
37     while k>0 loop
38         for j in 1..k loop
39             get(m); put(m,3);
40             setval(m);
41         end loop;
42         new_line; put("SUM =");
43         put(ArrayCalc.sum,4);
44         new_line; get(k);
45         end loop;
46 end;
```

Figure A.1. Ada example to sum an array.

```
1  import java.io.*;
2  class DataConvert {
3  public int convert(byte ch) {return ch-'0';}};
4  class DataStore extends DataConvert {
5  public void initial(int a)
6      {ci=0;
7       size=a;};
8  void save(int a)
9      {store[ci++]=a;};
10 int setprint() { ci=0; return size;};
11 int printval() { return store[ci++];};
12 int sum()
13     {int arrsum = 0;
14      for(ci=0;ci<size;ci++)arrsum=arrsum+store[ci];
15      return arrsum;};
16 private static int maxsize = 9;
17 int size; // Size of array
18 int ci; // Current index into array
19 int[] store = new int[maxsize];};
20 class sample {
21 public static void main(String argv[])
22     {int sz,j;
23      byte[] Line = new byte[10];
24      DataStore x = new DataStore();
25      try{
26          while((sz= System.in.read(Line)) != 0)
27              {int k = x.convert(Line[0]);
28               x.initial(k);
29               for(j=1;j<=k;j++)x.save(x.convert(Line[j]));
30               for(j=x.setprint(); j>0; j--)
31                   System.out.print(x.printval());
32               System.out.print("; SUM=");
33               System.out.println(x.sum());}}
34      catch(Exception e){System.out.println("File error.");}
35      } // End main
36      } // End class sample
```

Figure A.9. Java example to sum an array.


```
1 %lisp
2 >; Store values as a list of characters
3 >(define (SumNext V)
4     (cond ((null V) (progn (print "Sum=") 0))
5           (T (+ (SumNext (cdr V)) (car V)) ) ) )
6 SUMNEXT
7 >; Create vector of input values
8 (defun GetInput(f c)
9     (cond ((eq c 0) nil)
10          (T (cons (read f) (GetInput f (- c 1))))))
11 GETINPUT
12 >(defun DoIt()
13     (progn
14       (setq infile (open "lisp.data"))
15       (setq array (GetInput infile (read infile)))
16       (print array)
17       (print (SumNext array))))
18 DOIT
19 >(DoIt)
20
21 (1 2 3 4)
22 "Sum="
23 10
24 10
```

Figure A.10. Sum of a vector of numbers in LISP.

```
1 %editor data.prolog
2     /* Read in data as a Prolog Relation */
3     datavals(4,[1,2,3,4]).
4 %editor pgm.prolog
5     go :- reconsult('data.prolog'),
6           datavals(A,B),
7           INSUM is 0,
8           for(A,B,INSUM,OUTSUM),nl,
9           write('SUM ='),write(OUTSUM),nl.
10    /* for loop executes 'I' times */
11    for(I,B,INSUM,OUTSUM) :- not(I=0),
12      B=[HEAD|TAIL],
13      write(HEAD),
14      NEWVAL is INSUM+HEAD,
15      for(I-1,TAIL,NEWVAL,OUTSUM).
16    /* If I is 0, return INSUM computed value */
17    for(_,_ ,INSUM,OUTSUM) :- OUTSUM = INSUM.
18    not(X) :- X, !, fail.
19    not(_).
20 %prolog
21 | ?- consult('pgm.prolog').
22 {consulting /aaron/mvz/book/pgm.prolog...}
23 {/aaron/mvz/book/pgm.prolog consulted, 30 msec 1456 bytes}
24 yes
25 | ?- go.
26 {consulting /aaron/mvz/book/data.prolog...}
27 {/aaron/mvz/book/data.prolog consulted, 10 msec 384 bytes}
28 1234
29 SUM =10
30 yes
```

Figure A.16. Prolog example to sum an array.

```
1 # Read in and add up a vector of numbers.
2
3 def vector_sum(numbers):
4     total = 0.0
5     for number in numbers:
6         total += number
7
8     return total
9
10 alist = []           # A list of values, note, no size.
11
12 n = int( raw_input( "How many numbers are there " ) )
13
14 for i in range ( 1, n+1):
15     alist += [float(raw_input("Enter number %d: " % (i) ) ) ]
16
17 print "Sum is " , vector_sum(alist) , "\n"
18
```

python version