Lecture 3: Arrays

Charudatt Kadolkar

Arrays

Array is a regular arrangement of objects. In C programming, A declaration

```
int   marks[10];
```

creates 10 `int` type variables, puts them in adjacent memories.

- The *array* of 10 variables is called by name `marks` .
- Individual elements of array are `int` variables namely, `marks[0]`,`marks[1]`, ..., `marks[9]` .
- The *size* of this array is 10.

Arrays

Since individual elements are only variables, they can be used whereever variables can be used.

For example, We can assign values to these variables.

```
marks[5]   = 96;
marks[7]   = 27;
```

We can construct expressions with these variables

```
total  = marks[5]   + marks[7];
marks[8]   = marks[5]   / 100;
```

Arrays

Variable `marks[8]` is nineth variable in array. 8 (the integer enclosed in [and]) is the *index* of the variable in array or simply *array index*. Array index can be any integer expression. For example,

```
i = 5;
marks[i]   = 96;
marks[i+2]   = 27;
marks[i+2]   = marks[i+1]   + marks[i];
```

Arrays

What is the output?

```
int i;
int x[10];

i = 0;
while ( i < 10 ) {
   x[i] = i*i;
   i = i + 1;
}

printf("%d %d %d\n", x[1], x[5], x[9]);
```

Arrays

What is the output?

```
int  i;
int  x[10];

i = 0;
while  ( i < 10 ) {
   x[i] = i*i;
   i = i + 1;
}

i = 0;
while  ( i < 10 ) {
    printf("%d\n", x[i]);
    i = i + 2;
}
```

Examination Statistics

```
int  i,  total,   average,   n,  highest;
int  marks[10];

scanf("%d",   &n);                         //  Assume   that  n <=  10

/*  Read  marks   and  store   in  array   */
i = 0;
while  ( i < n ) {
   scanf("%d",   &marks[i];
   i = i + 1;
}

/*  Calculate   Average   */
i = 0;  total  = 0;
while  ( i < n ) {
    total  = total  + marks[i];
    i = i + 1;
}
average  = total  / n;
```

Examination Statistics

```
/* Separate below average */
i = 0;
while ( i < n ) {
   if ( marks[i] < average ) printf("Student %d is below average\n",i);
   else printf("Student %d is above average\n",i);
   i = i + 1;
}

/* Find Highest */
highest = marks[0];
i = 1; total = 0;
while ( i < n ) {
    if ( marks[i] > highest ) highest = marks[i];
    i = i + 1;
}
```

Polynomials

A polynomial of degree n is given by

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

We want to write a program that evaluates a polynomial at a given x. The coefficients of the polynomial are input in the beginning.

Polynomials

```
int i, n;
float x, a[10];

/* Input coefficients */
scanf("%d", &n);
i = 0;
while ( i <= n ) { scanf("%f", &a[i]); i = i + 1; }

/* Input x and evaluate */
scanf("%f", &x);
i = 1;
p = a[0];
while ( i <= n ) {
    p = p + a[i] * pow(x,i);
    i = i + 1;
}
printf("p(%f) = %f\n", x, p);
```

Arrays as Vectors

If $\overrightarrow{x} = (x_1, x_2, x_3)$ and $\overrightarrow{y} = (y_1, y_2, y_3)$ are two vectors in 3D, the angle between these is given by

$$\cos^{-1}\left(\frac{\overrightarrow{x} \cdot \overrightarrow{y}}{|\overrightarrow{x}|\,|\overrightarrow{y}|}\right)$$

Angle between vectors

```
float   x[3];
float   y[3];
float   sx,  sy,  sxy;
int     i;

scanf("%f%f%F",     &x[0],   &x[1],   &x[2]);
scanf("%f%f%F",     &y[0],   &y[1],   &y[2]);

sx  = 0; sy  = 0; sxy  = 0;
i = 0;
while  ( i < 3 )
{
    sx  = sx  + x[i]*x[i];
    sy  = sy  + y[i]*y[i];
    sxy  = sxy  + x[i]*y[i];
    i = i + 1;
}

angle  = acos(sxy   / sqrt(sx*sy));
```

Sorting

```
int i, j, k;
int a[10];

/* Read the list */
i = 0;
while ( i < 10 ) { scanf("%d", &a[i]); i = i + 1; }

/* Sort */
i = 0;
while ( i < 9 ) {
    j = i + 1;
    while ( j < 10 )  {
        if ( a[i] > a[j]  ){
            k = a[i];
            a[i] = a[j];
            a[j] = k;
        }
        j = j + 1;
    }
    i = i + 1;
}
```