

The void type



The **void** type can occur in *derived* types declarations.

Examples:

```
int function1(void); // function without arguments
```

```
void function2(int); // function returns no value
```

103

Arrays



An **array** is a collection of elements of the same type. Array type is a **derived type**.

Arrays elements can be:

- objects of basic type,
- objects of enumerated type,
- other arrays,
- objects of type defined by the user,
- pointers,
- etc.

104



Arrays

Declaring array :

type_of_element array_name[size];

Array elements are identified by an integer index written inside square brackets.

In C the index begins with **zero** always!

Examples:

int a[10];

char tab_z[20];

float Ftab[15];

Array elements:

a[0] a[i] a[k +1] a[9] tab_z[19] Ftab[14]

105

Example 3

```
# include <stdio.h>
/* program showing how to use arrays */

int main (void)
{
int i;
int tab [5];
for (i =0; i < 5; i++) tab [i] = 10*i; // entering data
for (i =0; i <5; i++)
printf("Element %d equals %d\n", i, tab[i]);
return 0;
}
```

106



Arrays initialization

Arrays are initialized with a brace-enclosed list of constant expressions, i.e. an initializer list.

Examples:

```
long t[4] = { 72896, 15, 6, 70800};  
float ta[4] = {1.2, 5.0}; //the last two elements = 0.0  
int tt[ ] = {3, 6, 19}; // definition of 3-element array
```

107



Character arrays

```
char text1 [20]; // definition of 20-element array
```

Initializing character arrays

```
char text1 [20] = {"Alps" };
```

```
char text1 [20] = { 'A', 'l', 'p', 's' };
```

```
char text2 [ ] = {"Tatra"}; // size: 6 elements
```

```
char text2 [ ] = { 'T', 'a', 't', 'r', 'a' }; // size: 5 elements
```

108

Example 14



Problem:

Write a program reading 10 numbers and storing them in an array, and then calculating the maximum number and printing it on the screen.

Discussion:

Input data: 10 numbers

Output data: maximum number

Algorithm:

- Read 10 numbers using "for" loop and save them in the array
- Calculate the maximum number:
 - assign to *max* variable the first number from the array
 - using "for" loop check sequentially all the array elements, and if any element is greater than *max*, assign it to the *max*.
- Print the result.

109

```
#include <stdio.h> /*Program calculates the maximum number */

int main() {
    int i;
    float max, tab[10];
    printf("Program calculating the maximum number");
    for(i=0; i<10; i++) {
        printf(" Enter %d number: ", i+1);
        scanf("%f", &tab[i]);
    }
    max=tab[0];
    for(i=1; i<10; i++)
        if(tab[i]>max) max=tab[i];
    printf(" The maximum number equals %10.5f\n", max);
    getchar();
    return 0;
}
```

Example 15

Write a program reading characters typed on a keyboard and calculating the number of occurrences for each digit, each white character and other characters.

111

Program

```
# include <stdio.h>
/* program calculating the digits, white characters and
other characters*/

int main ()
{
    int c, i, nwhite, nother;
    int ndigit [10];
    nwhite = nother =0;
    for ( i =0; i < 10; ++i) ndigit [i] = 0;
                                     //array elements set to zero
```

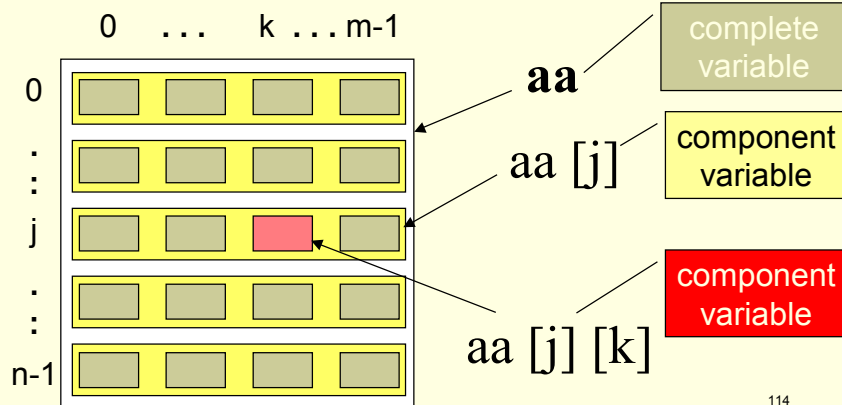
```
while ((c = getchar ()) != EOF)
    if (c >= '0' && c <= '9')
        ++ndigit [c-'0'];
    else if (c == ' ' || c == '\n' || c == '\t')
        ++nwhite;
    else ++nother;
printf("digits =");
for (i = 0; i < 10; ++i) printf("%d", ndigit[i]);
printf("\nwhite characters = %d, other = %d\n",
        nwhite, nother);

getchar();
return 0;
}
```

Multidimensional arrays



```
— const n=10, m=15;
int aa [n] [m]; //definition of two-dimensional array
int aaa [n] [10] [3]; // three-dimensional array
```



Initializing multidimensional arrays



Examples:

```
short aa[2][3] = {3, 7, 13, 8, 50, 71};
```

or

```
short aa[2][3] = {  
    {3, 7, 13},  
    {8, 50, 71}  
};
```

// group
// initialization

separator

terminator

are equivalent to:

```
short aa[2][3];  
aa[0][0] = 3; aa[0][1] = 7; aa[0][2] = 13;  
aa[1][0] = 8; aa[1][1] = 50; aa[1][2] = 71;
```

115

Initializing multidimensional arrays



Examples:

```
short aa[ ][3] = {  
    {3, 7, 13},  
    {8, 50, 71}  
};
```

correct

// correct

```
short aa[ ][3] = {3, 7, 13, 8, 50, 71};
```

// correct

```
int bbb[ ][2][2] = {0, 7, 8, 256, 90, 8, 675, 1};
```

correct

// correct

```
int bbb[ ] [ ] [2] = {0, 7, 8, 256, 90, 8, 675, 1};
```

error

error

116

Example 16

Write a program reading names and surnames of 15 students, storing them in an array and printing on the screen as follows:

- name surname
- or in a reversed order.

117

```
# include <stdio.h>
# include <stdlib.h>
/* program shows writing (reading) the data into (from)
   multidimensional array*/
int main (void)
{
    char c;
    char tab [15] [2] [20];
    int n,i;
    do {
        printf("Enter students number (n<16): ");
        i=scanf("%d", &n);
        fflush(stdin); //stdio.h - clearing the buffer
    } while (i==0);
    system("cls"); //stdlib.h – clearing the screen
    for (i =0; i < n; i++)
    {
```

118


```

printf("Enter name of %d student: ", i + 1 );
scanf("%19s", & tab [i] [0]); //writing the data into array
fflush(stdin);
printf("Enter surname of %d student: \n", i + 1);
scanf("%19s", & tab [i] [1]); //writing the data into array
fflush(stdin); printf(" \n");
}
printf("Printing firstly name, than surname? t/n");
fflush(stdin);
c=getchar();
printf("\n");
if (c=='t' || c=='T')
    for (i = 0 ; i <n; i++)
        printf("%d) %s %s\n", i+1, tab [i] [0], tab [i] [1]);
else for (i = 0 ; i <n; i++)
        printf("%d) %s %s\n", i+1, tab [i] [1], tab [i] [0]);
return 0;
}

```