**Лабораторна робота № 3. Внутрішнє подання базових структур даних**

***Мета:*** отримати та закріпити знання про внутрішнє (машинне) подання числових типів даних у мовах програмування.

**1 Вимоги**

**1.1 Розробник**

* Макаренко Владислав Олександрович
* Студент 1-го курсу
* Групи КІТ-120а

**1.2 Загальне завдання**

* Написати програму, яка виводить на екран внутрішнє (машинне) подання даних чотирьох типів. Типи даних обрати по таблиці 3.1 згідно із своїм номером у журналі групи. Тип элементів масиву обрати за своїм розсудом.

**1.3 Індивідуальні завдання**

* Типи: long int, double, char, array of char

**2 Описи програм**

**2.1 Внутрішнє подання типу:** long int

**2.1.1 Код програми**

#include <stdio.h>  
  
int main() {  
 long int n;  
 puts("Enter long int value: ");  
 scanf("%ld", &n);  
 int array[64] = {0};  
 int array1[8] = {0};  
 int array2[8] = {0};  
 int array3[8] = {0};  
 int array4[8] = {0};  
 int array5[8] = {0};  
 int array6[8] = {0};  
 int array7[8] = {0};  
 int array8[8] = {0};  
  
 for(int i = sizeof(n) \* 8 - 1; i >= 0; --i) {  
 array[i] = ((n >> i) & 1);  
 }  
 for(int i = 7; i >=0; i--){  
 array1[i] = array[i + 56];  
 }  
 for(int i = 7; i >=0; i--){  
 array2[i] = array[i + 48];  
 }  
 for(int i = 7; i >=0; i--){  
 array3[i] = array[i + 40];  
 }  
 for(int i = 7; i >=0; i--){  
 array4[i] = array[i + 32];  
 }  
 for(int i = 7; i >=0; i--){  
 array5[i] = array[i + 24];  
 }  
 for(int i = 7; i >=0; i--){  
 array6[i] = array[i + 16];  
 }  
 for(int i = 7; i >=0; i--){  
 array7[i] = array[i + 8];  
 }  
 for(int i = 7; i >=0; i--){  
 array8[i] = array[i];  
 }  
 for(int i = 7; i >=0; i--){  
 printf("%d", array8[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array7[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array6[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array5[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array4[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array3[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array2[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array1[i]);  
 }  
 printf(" ");  
 return 0;  
}

**2.1.2 Результати виконання програми**

Вводимо число *44444444454546845* і за алгоритмом коду дізнаємося його внутрішнє подання(див. рис. 1).

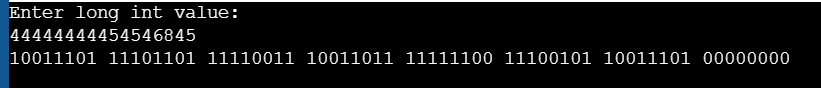


Рисунок 1 – Результати виконаного кроку

**2.2 Внутрішнє подання типу:** double

**2.2.1 Код програми**

#include <stdio.h>  
  
int main() {  
 double x;  
 puts("Enter double value: ");  
 scanf("%lf", &x);  
 long k = \*(long\*)&x;  
 int array[64] = {0};  
 int array1[8] = {0};  
 int array2[8] = {0};  
 int array3[8] = {0};  
 int array4[8] = {0};  
 int array5[8] = {0};  
 int array6[8] = {0};  
 int array7[8] = {0};  
 int array8[8] = {0};  
  
 for(int i = sizeof(k) \* 8 - 1; i >= 0; --i) {  
 array[i] = ((k >> i) & 1);  
 }  
 for(int i = 7; i >=0; i--){  
 array1[i] = array[i + 56];  
 }  
 for(int i = 7; i >=0; i--){  
 array2[i] = array[i + 48];  
 }  
 for(int i = 7; i >=0; i--){  
 array3[i] = array[i + 40];  
 }  
 for(int i = 7; i >=0; i--){  
 array4[i] = array[i + 32];  
 }  
 for(int i = 7; i >=0; i--){  
 array5[i] = array[i + 24];  
 }  
 for(int i = 7; i >=0; i--){  
 array6[i] = array[i + 16];  
 }  
 for(int i = 7; i >=0; i--){  
 array7[i] = array[i + 8];  
 }  
 for(int i = 7; i >=0; i--){  
 array8[i] = array[i];  
 }  
 for(int i = 7; i >=0; i--){  
 printf("%d", array8[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array7[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array6[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array5[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array4[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array3[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array2[i]);  
 }  
 printf(" ");  
 for(int i = 7; i >=0; i--){  
 printf("%d", array1[i]);  
 }  
 printf(" ");  
 return 0;  
}

**2.2.2 Результати виконання програми**

Вводимо число *15* і за алгоритмом коду дізнаємося його внутрішнє подання(див. рис. 2).

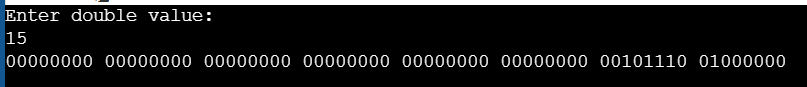
\ 

Рисунок 2 – Результати виконаного кроку

**2.3 Внутрішнє подання типу:** char

**2.3.1 Код програми**

int main()  
{  
 char symbol;  
 puts("Enter char value: ");  
 scanf("%c", &symbol);  
 for(int i = sizeof(symbol) \* 8 - 1; i >= 0; --i)  
 printf("%d", (symbol >> i) & 1);  
  
 return 0;  
}

**2.3.2 Результати виконання програми**

Вводимо символ V і за алгоритмом коду дізнаємося його внутрішнє подання(див. рис. 2).

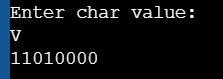


Рисунок 2 – Результати виконаного кроку

**2.3 Внутрішнє подання типу:** char

**2.3.1 Код програми**

int main()  
{  
 char symbol;  
 puts("Enter char value: ");  
 scanf("%c", &symbol);  
 for(int i = sizeof(symbol) \* 8 - 1; i >= 0; --i)  
 printf("%d", (symbol >> i) & 1);  
  
 return 0;  
}

**2.3.2 Результати виконання програми**

Вводимо символ V і за алгоритмом коду дізнаємося його внутрішнє подання(див. рис. 2).

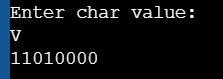


Рисунок 2 – Результати виконаного кроку

**2.4 Внутрішнє подання типу:**  array of char

**2.4.1 Код програми**

#include <stdio.h>  
  
#define SIZE 4  
  
int main() {  
 char symbols[SIZE] = {'v', 'l', 'a', 'd'};  
 for(int j = 0; j < SIZE; j++){  
 for(int i = sizeof(symbols[SIZE]) \* 8 - 1; i >= 0; --i){  
 printf("%d", (symbols[j] >> i) & 1);  
 }  
 printf("%c", ' ');  
 }  
 return 0;  
}

**2.4.2 Результати виконання програми**

Вводимо масив символів v, l, a, d і за алгоритмом коду дізнаємося його внутрішнє подання(див. рис. 2).



Рисунок 2 – Результати виконаного кроку

**Висновок:** на цій лабораторній роботі ми отримали та закріпили знання про внутрішнє (машинне) подання числових типів даних у мовах програмування.