|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт искусственного интеллекта |
| Кафедра Технологий искусственного интеллекта |

Практическая работа № 14

по дисциплине

«Процедурное программирование»

Обучающийся: Погосян С. А.

Группа: КВБО-07-23

Руководитель *Яковлев Д. А*

Москва 2023

**Тема: «Работа с указателями. Указатели на переменные, массивы, функции»**

**Цель работы: Исследовать принцип работы с указателями на функции и понять их область применения в практических задачах.**

**Описание работы:**

Написать функцию, которая возвращает несколько результатов. Кроме оператора return, для этого рекомендуется использовать указатели на переменные, в которых должны оказаться результаты работы функции. Объявить указатель на полученную функцию и вызвать её с помощью указателя.

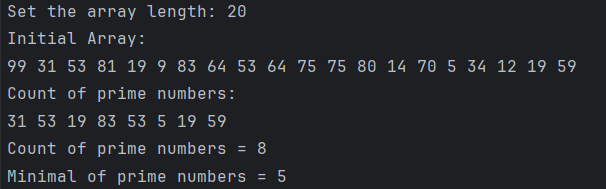
**Задача:**

****

**Код программы:**

#include <stdio.h>  
#include <time.h>  
#include <stdlib.h>  
  
int get\_count\_of\_prime\_numbers(int \*arr, int len) {  
 int count = 0; // Количество элементов  
 int count\_of\_div; // Количество делителей числа  
 printf("Count of prime numbers:\n");  
 for (int i = 0; i < len; i++) {  
 if (arr[i] != 1 && arr[i] != 0) {  
 count\_of\_div = 0;  
 for (int j = 1; j < arr[i]; j++) {  
 if (arr[i] % j == 0) {  
 count\_of\_div += 1;  
 if (count\_of\_div > 1) {  
 break;  
 }  
 }  
 }  
 if (count\_of\_div == 1) {  
 printf("%d ", arr[i]);  
 count += 1;  
 }  
 }  
 }  
 putchar('\n');  
 return count;  
  
}  
  
double get\_min\_prime\_number(int \*arr, int len, int \*mini) {  
 \*mini = 1000000;  
 int count\_of\_div;  
 int count\_of\_prime = get\_count\_of\_prime\_numbers(arr, len);  
 for (int i = 0; i < len; i++) {  
 if (arr[i] != 1 && arr[i] != 0) {  
 count\_of\_div = 0;  
 for (int j = 1; j < arr[i]; j++) {  
 if (arr[i] % j == 0) {  
 count\_of\_div += 1;  
 if (count\_of\_div > 1) {  
 break;  
 }  
  
 }  
 }  
 if (count\_of\_div == 1) {  
 if (\*mini > arr[i]) {  
 \*mini = arr[i];  
 }  
 }  
 }  
  
 }  
 return count\_of\_prime;  
}  
  
int main() {  
 srand(time(NULL));  
 int n;  
 printf("Set the array length: ");  
 scanf("%d", &n);  
 int \*arr = (int \*) malloc(sizeof(int) \* n);  
 // создаем указатель на функцию  
 double (\*fptr)(int \*, int, int \*) = NULL;  
 printf("Initial Array:\n");  
 for (int i = 0; i < n; i++) {  
 arr[i] = rand() % 100;  
 printf("%d ", arr[i]);  
 }  
 putchar('\n');  
 fptr = get\_min\_prime\_number;  
 int mini;  
 int get\_count\_of\_prime = fptr(arr, n, &mini);  
  
 printf("Count of prime numbers = %d\n", get\_count\_of\_prime);  
 printf("Minimal of prime numbers = %d\n", mini);  
  
 free(arr);  
  
 return 0;  
  
  
}

**Результат работы программы:**

****

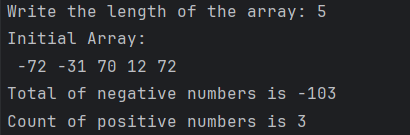
**Задание:**

****

**Код программы:**

#include <stdio.h>  
#include <time.h>  
#include <stdlib.h>  
  
  
int get\_total\_negative\_and\_count\_positive(int \*arr, int \*count\_p, int len) {  
 int total\_negative = 0;  
 for (int i = 0; i < len; i++) {  
 if (arr[i] < 0) {  
 total\_negative += arr[i];  
 } else if (arr[i] > 0) {  
 \*count\_p = \*count\_p + 1;  
 }  
 }  
 return total\_negative;  
}  
  
  
int main() {  
 srand(time(NULL));  
 int n;  
 int \*arr;  
 int count\_positive = 0;  
  
 printf("Write the length of the array: ");  
  
 scanf("%d", &n);  
  
 arr = (int \*) malloc(sizeof(int) \* n);  
  
 printf("Initial Array:\n ");  
  
 for (int i = 0; i < n; i++) {  
 arr[i] = rand() % (100 + 100 + 1) - 100;  
 printf("%d ", arr[i]);  
 }  
 putchar('\n');  
  
 int (\*gtncp)(int \*, int \*, int) = NULL;  
  
 gtncp = get\_total\_negative\_and\_count\_positive;  
  
 int get\_total\_negative = gtncp(arr, &count\_positive, n);  
  
 printf("Total of negative numbers is %d\n", get\_total\_negative);  
 printf("Count of positive numbers is %d\n", count\_positive);  
  
 free(arr);  
  
 return 0;  
}

**Результат работы программы:**

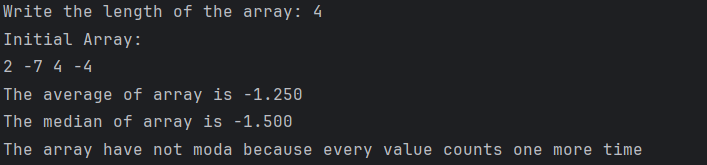
****

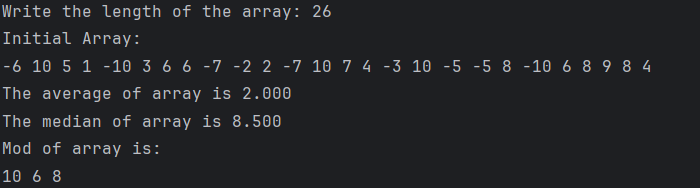
**Задание:**

****

#include <stdio.h>  
#include <time.h>  
#include <stdlib.h>  
  
int get\_avg\_median\_moda(int \*arr, double \*avg, double \*median, int \*moda, int \*count\_numbers\_of\_moda, int n) {  
  
 // Нахождение среднего арифметического ряда  
 int total = 0;  
 for (int i = 0; i < n; i++) {  
 total += arr[i];  
 }  
 \*avg = (double) (total) / (double) (n);  
  
  
 // Нахождение медианы ряда  
  
 if (n % 2 != 0) {  
 \*median = arr[n / 2];  
 } else {  
 \*median = (double)(arr[n / 2] + arr[n / 2 - 1]) / 2;  
 }  
  
 // Нахождение моды ряда  
  
 int max\_count = 1, cur\_count;  
 for (int i = 0; i < n; i++) {  
 cur\_count = 1;  
 for (int j = 0; j != i && j < n; j++) {  
 if (arr[i] == arr[j]) {  
 cur\_count += 1;  
 }  
 }  
 if (cur\_count > max\_count) {  
 max\_count = cur\_count;  
 }  
 }  
 if (max\_count != 1) {  
 int extra\_j = 0; // Текущее значение, которое повторяется максимальное количество раз  
 for (int i = 0; i < n; i++) {  
 cur\_count = 1;  
 for (int j = 0; j != i && j < n; j++) {  
 if (arr[i] == arr[j]) {  
 cur\_count += 1;  
 }  
 }  
 if (cur\_count == max\_count && arr[i] != extra\_j) {  
 moda[\*count\_numbers\_of\_moda] = arr[i];  
 \*count\_numbers\_of\_moda = \*count\_numbers\_of\_moda + 1;  
 extra\_j = arr[i];  
  
 }  
 }  
 moda = realloc(moda, \*count\_numbers\_of\_moda);  
 } else {  
 moda = NULL;  
 }  
 return 0;  
}  
  
  
int main() {  
 srand(time(NULL));  
 int n;  
 int \*arr;  
 double avg, median;  
 int \*moda;  
 int count\_numbers\_of\_moda = 0;  
  
 printf("Write the length of the array: ");  
 scanf("%d", &n);  
  
 arr = (int \*) malloc(sizeof(int) \* n);  
  
 moda = (int \*) malloc(sizeof(n) \* n);  
  
 printf("Initial Array:\n");  
  
  
 for (int i = 0; i < n; i++) {  
 arr[i] = rand() % (10 + 10 + 1) - 10;  
 printf("%d ", arr[i]);  
 }  
 putchar('\n');  
  
 int (\*gamm)(int \*, double \*, double \*, int \*, int \*, int);  
  
 gamm = get\_avg\_median\_moda;  
  
 gamm(arr, &avg, &median, moda, &count\_numbers\_of\_moda, n);  
  
 printf("The average of array is %.3lf\n", avg);  
 printf("The median of array is %.3lf\n", median);  
  
  
 if (count\_numbers\_of\_moda == 0) {  
 printf("The array have not moda because every value counts one more time\n");  
 } else {  
 printf("Mod of array is:\n");  
  
 for (int i = 0; i < count\_numbers\_of\_moda; i++) {  
 printf("%d ", moda[i]);  
 }  
 }  
  
  
 return 0;  
}

**Результат работы программы:**

****

****