```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

#define STRING_LENGTH 80

int readLine(char str[], int numberOfCharacters) {
  int currentChar = 0, currentNumber = 0;

  while ((currentChar = getchar()) != '\n' && currentChar != EOF) {
    if (currentNumber < numberOfCharacters) {
      str[currentNumber++] = currentChar;
    }
  }

  str[currentNumber] = '\0';

  return currentNumber;
}

int main(void) {
  setlocale(LC_ALL, "RU");

  char string[STRING_LENGTH + 1] = {'\0'};

  printf("Введите текст:\n");

  readLine(string, STRING_LENGTH);

  puts("");

  puts(string);

  return EXIT_SUCCESS;
}
```

```
int countWhitespaces(const char str[]) {
  int count = 0;

  for (int i = 0; str[i] != '\0'; i++) {
    if (str[i] == ' ') {
      count++;
    }
  }

  return count;
}
```

```
int countWhitespaces(const char *str) {
  int count = 0;

  for (; *str != '\0'; str++) {
    if (*str == ' ') {
      count++;
    }
  }

  return count;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

#define SIZE 10

int main(void) {
  setlocale(LC_ALL, "RU");

  char firstString[SIZE], secondString[SIZE];

  firstString = "abc";

  secondString = firstString;

  printf("%d\n", firstString == secondString);

  return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>

#define SIZE 10

int main(void) {
  setlocale(LC_ALL, "RU");

  char firstString[SIZE], secondString[SIZE];
  // char *strcpy( char *dest, const char *src );
  strcpy(secondString, "abcd");

  strcpy(firstString, secondString);

  strcpy(firstString, strcpy(secondString, "abcd"));
  //char *strncpy( char *dest, const char *src, size_t count );
  strncpy(firstString, secondString, sizeof(firstString) - 1);

  firstString[sizeof(firstString) - 1] = '\0';

  return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>

#define SIZE 10

int main(void) {
  setlocale(LC_ALL, "RU");

  char firstString[SIZE];
  // size_t strlen( const char *str );
  int lenght = strlen("abc def");
  printf("%d\n", lenght);

  lenght = strlen("");
  printf("%d\n", lenght);

  strcpy(firstString, "abc");

  lenght = strlen(firstString);
  printf("%d\n", lenght);

  return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>

#define SIZE 10

int main(void) {
  setlocale(LC_ALL, "RU");

  char firstString[SIZE], secondString[SIZE];

  strcpy(firstString, "abc");

  // char *strcat( char *dest, const char *src );
  strcat(firstString, "def");

  //strcpy(secondString, "def");
  //strcat(firstString, secondString);
  //strncat(firstString, secondString,
  //        sizeof(firstString) - strlen(firstString) - 1);
  //firstString[sizeof(firstString) - 1] = '\0';

  puts(firstString);
  return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>

#define SIZE 10

int main(void) {
  setlocale(LC_ALL, "RU");

  char firstString[SIZE], secondString[SIZE];

  strcpy(firstString, "abc");

  strcpy(secondString, "def");

  // int strcmp( const char *lhs, const char *rhs );

  printf("%d\n", strcmp(firstString, secondString));

  return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>

#define SIZE 10

size_t GetStringLength(const char *s) {
  size_t n;

  for (n = 0; *s != '\0'; s++) {
    n++;
  }

  return n;
}

int main(void) {
  setlocale(LC_ALL, "RU");

  char firstString[SIZE], secondString[SIZE];

  strcpy(firstString, "abc");

  strcpy(secondString, "def");

  printf("%d\n", GetStringLength(firstString));

  return EXIT_SUCCESS;
}
```

```c
  while (*s++) {
    n++;
  }
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>

#define SIZE 10

char* ConcatinateStrings(char *first, const char *second) {
  char *p = first;
  while (*p != '\0') {
    p++;
  }
  while (*second != '\0') {
    *p = *second;
    p++;
    second++;
  }
  *p = '\0';
  return first;
}

int main(void) {
  setlocale(LC_ALL, "RU");
  char firstString[SIZE], secondString[SIZE];
  strcpy(firstString, "abc");
  strcpy(secondString, "def");
  printf("%s\n", ConcatinateStrings(firstString, secondString));
  return EXIT_SUCCESS;
}
```

```c
  while (*p) {
    p++;
  }

  while (*p++ = *second++)
    ;
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

#define SIZE 10

int main(void) {
  setlocale(LC_ALL, "RU");

  char planets[][SIZE] = { "Меркурий", "Венера",
                           "Земля", "Марс",
                           "Юпитер", "Сатурн",
                           "Уран", "Нептун" };

  return EXIT_SUCCESS;
}
```

```c
  char *planets[] = { "Меркурий", "Венера",
                      "Земля", "Марс",
                      "Юпитер", "Сатурн",
                      "Уран", "Нептун" };
```

```c
  int size = sizeof(planets) / sizeof(char*);

  for (int i = 0; i < size; i++) {
    if (planets[i][0] == 'M') {
      printf("%s пишется с буквы M\n", planets[i]);
    }
  }
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

int main(int argc, char *argv[]) {
  setlocale(LC_ALL, "RU");

  for (int i = 1; i < argc; i++) {
    printf("%s\n", argv[i]);
  }

  return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

int main(int argc, char *argv[]) {
  setlocale(LC_ALL, "RU");

  for (char **p = &argv[1]; *p != NULL; p++) {
    printf("%s\n", *p);
  }

  return EXIT_SUCCESS;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <string.h>

int main(int argc, char *argv[]) {
  setlocale(LC_ALL, "RU");

  char *planets[] = { "Меркурий", "Венера",
                      "Земля", "Марс",
                      "Юпитер", "Сатурн",
                      "Уран", "Нептун" };

  int numberOfPlanets = sizeof(planets) / sizeof(char*);

  int i = 0, j = 0;

  for (i = 1; i < argc; i++) {
    for (j = 0; j < numberOfPlanets; j++) {
      if (strcmp(argv[i], planets[j]) == 0) {
        printf("%s это планета № %d\n", argv[i], j + 1);
        break;
      }
    }
    if (j == numberOfPlanets) {
      printf("%s это не планета\n", argv[i]);
    }
  }

  return EXIT_SUCCESS;
}
```

Некоторые функции библиотеки **ctype.h**

```
int isdigit( int ch );
int isalpha( int ch );
int isalnum( int ch );
int isxdigit( int ch );
int islower( int ch );
int isupper( int ch );
int tolower( int ch );
int toupper( int ch );
int isspace( int ch );
int iscntrl( int ch );
int ispunct( int ch );
int isprint( int ch );
int isgraph( int ch );
```

Некоторые функции библиотеки **stdlib.h**

```
double strtod( const char *str,
               char **str_end );

long strtol( const char *str,
             char **str_end, int base );

unsigned long strtoul( const char *str,
                       char **str_end,
                       int base );
```

Некоторые функции поиска из библиотеки **`string.h`**

```
char* strchr( const char *str, int ch );

size_t strspn( const char *dest,
               const char *src );

char* strpbrk( const char *dest,
               const char *breakset );
```