C Inbuilt Functions

1. **`scanf()` and `printf()`**
   - **Purpose**: Input and output operations.
   - **Example**:

```c
int number;
printf("Enter a number: ");
scanf("%d", &number);
printf("You entered: %d\n", number);
```

2. **`strlen()`**
   - **Purpose**: Compute the length of a string.
   - **Example**:

```c
#include <string.h>
char str[] = "Hello";
int length = strlen(str);
printf("Length of string: %d\n", length);
```

3. **`strcpy()` and `strcat()`**
   - **Purpose**: Copy and concatenate strings.
   - **Example**:

```c
#include <string.h>
char dest[20], src[] = "Hello";
strcpy(dest, src);
printf("Copied string: %s\n", dest);
strcat(dest, " World");
printf("Concatenated string: %s\n", dest);
```

4. **`isdigit()` and `isalpha()`**
   - **Purpose**: Check if a character is a digit or alphabetic.
   - **Example**:

```c
#include <ctype.h>
char ch = '7';
if (isdigit(ch)) {
    printf("%c is a digit\n", ch);
}
```

5. `atoi()` **and** `atof()`

   - **Purpose**: Convert strings to integers or floating-point numbers.
   - **Example**:

```c
#include <stdlib.h>
char numStr[] = "123";
int num = atoi(numStr);
printf("Converted number: %d\n", num);
```

6. `abs()`

   - **Purpose**: Compute the absolute value of an integer.
   - **Example**:

```c
#include <stdlib.h>
int x = -10;
int absX = abs(x);
printf("Absolute value of %d is %d\n", x, absX);
```

7. `pow()` **and** `sqrt()`

   - **Purpose**: Perform exponentiation and square root calculations.
   - **Example**:

```c
#include <math.h>
double result1 = pow(2.0, 3.0);  // 2^3 = 8
double result2 = sqrt(16.0);     // Square root of 16 = 4
```

8. `strcmp()`

   - **Purpose**: Compare two strings.
   - **Example**:

```c
#include <string.h>
char str1[] = "hello";
char str2[] = "world";
int result = strcmp(str1, str2);
if (result == 0) {
    printf("Strings are equal\n");
} else {
    printf("Strings are not equal\n");
}
```

9. **`scanf()` with `%[^\n]` format specifier**
    - **Purpose**: Read a line of text including spaces.
    - **Example**:

```c
char sentence[100];
printf("Enter a sentence: ");
scanf(" %[^\n]", sentence);
printf("You entered: %s\n", sentence);
```

10. **`strtok()`**
    - **Purpose**: Tokenize a string based on delimiters.
    - **Example**:

```c
#include <string.h>
char sentence[] = "Hello, world!";
char *token = strtok(sentence, ", ");
while (token != NULL) {
    printf("Token: %s\n", token);
    token = strtok(NULL, ", ");
}
```

11. **`fgets()`**
    - **Purpose**: Read a line of input from a file stream (including stdin).
    - **Example**:

```c
#include <stdio.h>
char buffer[100];
```

```
printf("Enter a string: ");
fgets(buffer, sizeof(buffer), stdin);
printf("You entered: %s", buffer);
```

12. **`strstr()`**

- **Purpose**: Find the first occurrence of a substring within a string.
- **Example**:

```
#include <string.h>
char str[] = "Hello, world!";
char *substr = "world";
char *result = strstr(str, substr);
if (result != NULL) {
    printf("Substring found at position: %ld\n", result -
} else {
    printf("Substring not found\n");
}
```

13. **`tolower()` and `toupper()`**

- **Purpose**: Convert characters to lowercase or uppercase.
- **Example**:

```
#include <ctype.h>
char ch = 'A';
char lowerCh = tolower(ch);
char upperCh = toupper(ch);
printf("Lowercase: %c, Uppercase: %c\n", lowerCh, upperCh
```

14. **`fgets()` with `sscanf()` for parsing**

- **Purpose**: Read input as a string and parse formatted data from it.
- **Example**:

```
#include <stdio.h>
char buffer[100];
int num;
printf("Enter a number: ");
```

```
fgets(buffer, sizeof(buffer), stdin);
sscanf(buffer, "%d", &num);
printf("Parsed number: %d\n", num);
```

15. **`isalnum()`**

    - **Purpose**: Check if a character is alphanumeric.
    - **Example**:

```
#include <ctype.h>
char ch = 'A';
if (isalnum(ch)) {
    printf("%c is alphanumeric\n", ch);
} else {
    printf("%c is not alphanumeric\n", ch);
}
```

16. **`strncpy()`**

    - **Purpose**: Copy a specified number of characters from one string to another.
    - **Example**:

```
#include <string.h>
char src[] = "Hello";
char dest[10];
strncpy(dest, src, sizeof(dest) - 1);
dest[sizeof(dest) - 1] = '\0';  // Ensure null-terminated
printf("Copied string: %s\n", dest);
```

17. **`strchr()` and `strrchr()`**

    - **Purpose**: Find the first or last occurrence of a character in a string.
    - **Example**:

```
#include <string.h>
char str[] = "Hello, world!";
char ch = 'o';
char *first = strchr(str, ch);
```

```
char *last = strrchr(str, ch);
printf("First occurrence at position: %ld\n", first - str
printf("Last occurrence at position: %ld\n", last - str);
```

18. **`sprintf()`**

    - **Purpose**: Write formatted data to a string.
    - **Example**:

```
#include <stdio.h>
char buffer[100];
int num = 123;
sprintf(buffer, "Number: %d", num);
printf("Formatted string: %s\n", buffer);
```

19. **`strtol()` and `strtod()`**

    - **Purpose**: Convert strings to long integers or floating-point numbers with error checking.
    - **Example**:

```
#include <stdlib.h>
char numStr[] = "123";
long intVal = strtol(numStr, NULL, 10);
double dblVal = strtod(numStr, NULL);
printf("Converted integer: %ld, Converted double: %.2f\n"
```

20. **`memcpy()`**

    - **Purpose**: Copy a block of memory from one location to another.
    - **Example**:

```
#include <string.h>
char src[] = "Hello";
char dest[10];
memcpy(dest, src, strlen(src) + 1);
printf("Copied string: %s\n", dest);
```

Certainly! Here are more useful C standard library functions from `ctype.h`, `string.h`, and `stdlib.h` that are commonly used for problem-solving tasks:

From `<ctype.h>`:

21. **`isupper()` and `islower()`**

- **Purpose**: Check if a character is uppercase or lowercase.
- **Example**:

```c
#include <ctype.h>
char ch = 'A';
if (isupper(ch)) {
    printf("%c is uppercase\n", ch);
} else if (islower(ch)) {
    printf("%c is lowercase\n", ch);
}
```

22. **`isxdigit()`**

- **Purpose**: Check if a character is a hexadecimal digit (0-9, A-F, a-f).
- **Example**:

```c
#include <ctype.h>
char ch = 'F';
if (isxdigit(ch)) {
    printf("%c is a hexadecimal digit\n", ch);
}
```

23. **`isspace()`**

- **Purpose**: Check if a character is a whitespace character.
- **Example**:

```c
#include <ctype.h>
char ch = ' ';
if (isspace(ch)) {
```

```
    printf("%c is a whitespace character\n", ch);
}
```

24. **`tolower()` and `toupper()`**

- **Purpose**: Convert characters to lowercase or uppercase.
- **Example**:

```
#include <ctype.h>
char ch = 'a';
char upperCh = toupper(ch);
printf("Uppercase: %c\n", upperCh);
```

From `<string.h>`:

25. **`strncat()`**

- **Purpose**: Concatenate a specified number of characters from one string to another.
- **Example**:

```
#include <string.h>
char dest[20] = "Hello";
char src[] = " World!";
strncat(dest, src, 6);  // Concatenate first 6 characters from
printf("Concatenated string: %s\n", dest);
```

26. **`strncmp()`**

- **Purpose**: Compare the first `n` characters of two strings.
- **Example**:

```
#include <string.h>
char str1[] = "Hello";
char str2[] = "Hella";
int result = strncmp(str1, str2, 4);  // Compare first 4 chara
if (result == 0) {
    printf("First 4 characters are the same\n");
```

```
} else {
    printf("First 4 characters are different\n");
}
```

27. `strspn()`

- **Purpose**: Get the length of the initial segment of a string that contains only characters from a specified set.
- **Example**:

```
#include <string.h>
char str[] = "123abc456";
char charset[] = "0123456789";
size_t len = strspn(str, charset);
printf("Length of initial segment: %zu\n", len);
```

28. `strtok()`

- **Purpose**: Tokenize a string into smaller strings (tokens) based on delimiters.
- **Example**:

```
#include <string.h>
char str[] = "Hello,World,Everyone";
char *token = strtok(str, ",");
while (token != NULL) {
    printf("Token: %s\n", token);
    token = strtok(NULL, ",");
}
```

From `<stdlib.h>`:

29. `rand()` and `srand()`

- **Purpose**: Generate pseudorandom numbers.
- **Example**:

```c
#include <stdlib.h>
#include <time.h>
srand(time(NULL));  // Initialize random seed
int randomNum = rand() % 100;  // Generate random number betwe
printf("Random number: %d\n", randomNum);
```

30. **`qsort()`**

- **Purpose**: Sort an array using a specified comparison function.
- **Example**:

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

int compare(const void *a, const void *b) {
    return (*(int *)a - *(int *)b);
}

int main() {
    int arr[] = {5, 2, 8, 1, 6};
    int n = sizeof(arr) / sizeof(arr[0]);
    qsort(arr, n, sizeof(int), compare);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
printf("\n");
    return 0;
}
```