Experiment 1: Substring Extraction

Problem Statement

Write a C program to extract a substring from a given string based on position and length.

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>
int main() {
    char str[100], sstr[100];
    int pos, l, c = 0;

    printf("\n\nExtract a substring from a given string:\n");
    printf("Input the string : ");
    fgets(str, sizeof str, stdin);
    printf("Input the position to start extraction :");
    scanf("%d", &pos);
    printf("Input the length of substring :");
    scanf("%d", &l);
    while (c < l) {
        sstr[c] = str[pos + c - 1];
        c++;
    }
    sstr[c] = '\0';
    printf("The substring retrieved from the string is : \" %s\" \n\n", sstr);
    return 0;
}
```

Sample Output

Input:
Hello World
Position: 1
Length: 5
Output:

The substring retrieved from the string is : "Hello"

Experiment 2: String Comparison

Problem Statement

Write a C program to compare two strings using strcmp() function.

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>
#include <string.h>

int main() {
  char str1[] = "Hello";
  char str2[] = "Hello";
  char str3[] = "Hi";

  printf("%d\n", strcmp(str1, str2));
  printf("%d\n", strcmp(str1, str3));

  return 0;
}
```

Sample Output

Output:
0
-1

Experiment 3: String Copy

Problem Statement

Write a C program to copy one string into another using strcpy().

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>
#include <string.h>

int main() {
  char str1[20] = "Hello World!";
  char str2[20];
  strcpy(str2, str1);
  printf("%s", str2);
  return 0;
}
```

Sample Output

Output:
Hello World!

Experiment 4: String Concatenation

Problem Statement

Write a C program to concatenate two strings using strcat().

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>
#include <string.h>

int main() {
  char str1[20] = "Hello ";
  char str2[] = "World!";
  strcat(str1, str2);
  printf("%s", str1);
  return 0;
}
```

Sample Output

Output:
Hello World!

Experiment 5: String Length

Problem Statement

Write a C program to find the length of a string using strlen().

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>
#include <string.h>

int main() {
  char alphabet[50] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  printf("Length is: %d\n", strlen(alphabet));
  printf("Size is: %d\n", sizeof(alphabet));
  return 0;
}
```

Sample Output

Output:
Length is: 26
Size is: 50

Experiment 6: Character in String

Problem Statement

Write a C program to find the first occurrence of a character in a string using strchr().

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>
#include <string.h>

int main() {
    char str[] = "GeeksforGeeks";
    char ch = 'e';
    char* result = strchr(str, ch);
    if (result != NULL) {
        printf("The character '%c' is found at index %ld\n", ch, result -
str);
    } else {
        printf("The character '%c' is not found in the string\n", ch);
    }
    return 0;
}
```

Sample Output

Output:

The character 'e' is found at index 1

Experiment 7: First Occurrence of Substring

Problem Statement

Write a C program to find the first occurrence of a substring in a string using strstr().

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>
#include <string.h>

int main() {
    char s1[] = "GeeksforGeeks";
    char s2[] = "for";
    char* result;
    result = strstr(s1, s2);
    if (result != NULL) {
        printf("Substring found: %s\n", result);
    } else {
        printf("Substring not found.\n");
    }
    return 0;
}
```

Sample Output

Output:

Substring found: forGeeks

Experiment 8: String Index Search

Problem Statement

Write a C program to check whether a substring is present in a given string and find its index.

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>

int main() {
    char str[80], search[20];
    int c1 = 0, c2 = 0, i, j, flg, pos=0;

    printf("Input the string : ");
    fgets(str, sizeof str, stdin);
    printf("Input the substring to be searched : ");
    fgets(search, sizeof search, stdin);

    while (str[c1] != '\0') c1++;
    c1--;
    while (search[c2] != '\0') c2++;
    c2--;

    for (i = 0; i <= c1 - c2; i++) {
        for (j = i; j < i + c2; j++) {
            flg = 1;
            if (str[j] != search[j - i]) {
                flg = 0;
                break;
            }
        }
        if (flg == 1) { pos=i; break; }
    }
```

```
    if (flg == 1)
        printf("The substring exists in the string at %d.\n\n", pos);
    else
        printf("The substring does not exist in the string.\n\n");

    return 0;
}
```

Sample Output

Output:
The substring exists in the string at 5.

Experiment 9: Insert Substring

Problem Statement

Write a C program to insert one string into another at a specified position.

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

```
#include <stdio.h>
#include <string.h>

int main() {
   char str[80], add[20];
   int l1 = 0, l2 = 0, i, n;

   printf("Input the string : ");
   fgets(str, sizeof str, stdin);
   printf("Input the substring to be added : ");
   fgets(add, sizeof add, stdin);
   while (str[l1] != '\0') l1++;
```

```
    l1--;
    while (add[l2] != '\0') l2++;
    l2--;

    printf("Enter the position where the string is to be inserted\n");
    scanf("%d", &n);
    for(i = l1; i >= n; i--)
        str[i + l2] = str[i];

    for(i = 0; i < l2; i++)
        str[n + i] = add[i];

    str[l1 + l2] = '\0';
    printf("After inserting the string is %s", str);
    return 0;
}
```

Sample Output

Input:
String: HelloWorld
Insert: _C_
Position: 5
Output:
Hello_C_World

Experiment 10: Delete Substring

Problem Statement

Write a C program to delete a portion of a string starting from a specific position.

Objective

To understand string manipulation operations in C using standard library functions.

Algorithm

1. Start the program.
2. Declare the necessary string variables.
3. Take input from the user as required.
4. Perform string operation (extract, compare, copy, concatenate, etc.).
5. Display the result.
6. Stop the program.

Program Code

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

int main() {
   char str[80];
   int l = 0, i, n, pos;

   printf("Input the string : ");
   fgets(str, sizeof str, stdin);
   printf("Enter the position where the characters are to be deleted\n");
   scanf("%d", &pos);
   printf("Enter the number of characters to be deleted\n");
   scanf("%d", &n);

   l = strlen(str);
   for(i = pos + n; i < l; i++)
      str[i - n] = str[i];
   str[i - n] = '\0';

   printf("The string is %s", str);
   return 0;
}
```

Sample Output

Input:
HelloWorld
Position: 5
Delete: 3
Output:
Helloorld