

Factorial of a number using function and pointer variables in c

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

```
void findFactorial(int,int *);
```

```
int main(){
```

```
int i,factorial,num;
```

```
printf("Enter a number: ");
```

```
scanf("%d",&num);
```

```
findFactorial(num,&factorial);
```

```
printf("Factorial of %d is: %d",num,*factorial);
```

```
return 0;
```

```
}
```

```
void findFactorial(int num,int *factorial){
```

```
int i;
```

```
*factorial =1;
```

```
for(i=1;i<=num;i++)
```

```
*factorial=*factorial*i;
```

```
}
```

Display alphabets from A to Z or a to z

```
#include <stdio.h>
int main() {
    char c;
    printf("Enter u to display uppercase alphabets.
\n");
    printf("Enter l to display lowercase alphabets.
\n");
    scanf("%c", &c);

    if (c == 'U' || c == 'u') {
        for (c = 'A'; c <= 'Z'; ++c)
            printf("%c ", c);
    } else if (c == 'L' || c == 'l') {
        for (c = 'a'; c <= 'z'; ++c)
            printf("%c ", c);
    } else {
        printf("Error! You entered an invalid
character.");
    }
}
```

```
    return 0;
}
```

Program to Check whether a character is Alphabet

```
#include <stdio.h>
int main() {
    char c;
    printf("Enter a character: ");
    scanf("%c", &c);

    if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'))
        printf("%c is an alphabet.", c);
    else
        printf("%c is not an alphabet.", c);

    return 0;
}
```

Remove all characters in a string except alphabets

```
#include <stdio.h>
int main() {
    char line[150];

    printf("Enter a string: ");
    fgets(line, sizeof(line), stdin); //
take input

    for (int i = 0, j; line[i] != '\0'; ++i) {

        // enter the loop if the character
is not an alphabet
        // and not the null character
```

```
while (!(line[i] >= 'a' && line[i] <=
'z') && !(line[i] >= 'A' && line[i] <= 'Z')
&& !(line[i] == '\0')) {
    for (j = i; line[j] != '\0'; ++j) {

        // if jth element of line is not
an alphabet,
        // assign the value of (j+1)th
element to the jth element
        line[j] = line[j + 1];
    }
    line[j] = '\0';
}
}
printf("Output String: ");
puts(line);
return 0;
}
```

