

② Pattern matching :-

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

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

```
int string-m (char t[], char p[]) {
```

```
    int n = strlen(t);
```

```
    int m = strlen(p);
```

```
    for (int i = 0; i <= (n-m); i++) {
```

```
        int j = 0;
```

```
        while (j < m & t[i+j] == p[j]) {
```

```
            j++;
```

```
        }
```

```
        if (j == m) {
```

```
            return i;
```

```
        }
```

```
    }
```

```
int main () {
```

```
    char t[100], p[100];
```

```
    printf ("Enter the text : ");
```

```
    scanf ("%s", t);
```

```
    printf ("Enter the pattern : ");
```

```
    scanf ("%s", p);
```

```
    int result = string-m(t, p);
```

```
    if (result != -1) {
```

```
        printf ("Pattern found at index %d\n", result + 1);
```

```
    } else {
```

```

    printf("Pattern not found\n");
}
return 0;

```

<N.oid.12> should be
 <N.girt> should be

```

{ (C) & nolo, (C) & nolo } m - girt. b.
{ (T) & nolo, (T) & nolo } m - girt. b.
{ (T) & nolo, (T) & nolo } m - girt. b.

```

Output :-

Enter the text = funworld
 Enter the pattern = world
 Pattern found at index 4.

```

{ (T) & nolo, (T) & nolo } m - girt. b.
{ (T) & nolo, (T) & nolo } m - girt. b.
{ (T) & nolo, (T) & nolo } m - girt. b.

```

```

{ (C) & nolo, (C) & nolo } m - girt. b.
{ (C) & nolo, (C) & nolo } m - girt. b.
{ (C) & nolo, (C) & nolo } m - girt. b.

```

N
 13/6/24

```

{ (m) & nolo, (m) & nolo } m - girt. b.

```

i m m m m m

{ (C) & nolo, (C) & nolo } m - girt. b.

```

{ (C) & nolo, (C) & nolo } m - girt. b.

```

printf("Enter the text")

{ (T) & nolo, (T) & nolo } m - girt. b.

printf("Enter the pattern")

{ (T) & nolo, (T) & nolo } m - girt. b.

{ (T) & nolo, (T) & nolo } m - girt. b.

{ (T) & nolo, (T) & nolo } m - girt. b.

printf("Pattern found at index %d", i);

return 0;