

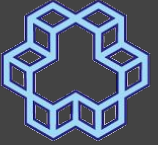


Fundamentals of Programming

session 28

Formatted Streams

```
*****  
* convolve.c  
***** /  
  
/* Standard includes */  
#include <assert.h>  
#include <math.h>  
#include <stdlib.h> /* malloc(), realloc() */  
  
/* Our includes */  
#include "base.h"  
#include "error.h"  
#include "convolve.h"  
#include "klt_util.h" /* printing */  
  
#define MAX_KERNEL_WIDTH 71  
  
typedef struct {  
    int toth;  
    float data[MAX_KERNEL_WIDTH];  
} ConvolutionKernel;  
  
/* Kernels */
```



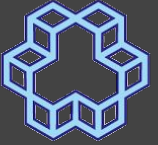
1926

K. N. T. University of Technology

Streams

- A sequence of data (bytes)
 - processed sequentially
 - potentially unlimited
- Stream vs. Batch data
- user input, user output, files, I/O devices, etc.

Stream Processing



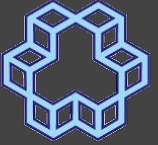
1926

K. J. Somaiya Institute of Technology

423423234532asdfsadfasdfzxc...

```
while (1) {  
    char c = getchar();  
    putchar(toupper(c));  
}
```

423423234532ASDFSADFSDFZXC...



1926

K. J. Somaiya Institute of Technology

Standard streams

- **Standard Input**
 - **Standard Output**
 - **Standard Error**
-
- **Redirection**
 - **Piping**



1926

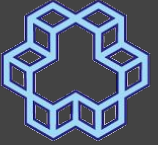
K. J. Somaiya Institute of Technology

Standard streams

- **Standard Input**
 - **Standard Output**
 - **Standard Error**
-
- **Redirection**
 - **Piping**

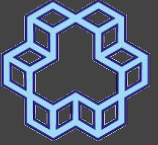
Printf

```
printf(control-string, args...)
```



1926

K. J. Somaiya Institute of Technology

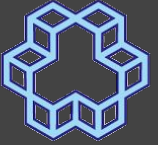


1926

K. N. Toosi University of Technology

formatting integers

<code>%d, %i</code>	signed integers (base 10)
<code>%u</code>	unsigned integers (base 10)
<code>%o</code>	octal unsigned integers (base 8)
<code>%x, %X</code>	hexadecimal unsigned integers (base 16)
<code>h short</code>	<code>%hd, %hu, %hx</code>
<code>l long</code>	<code>%ld, %lu, %lx</code>
<code>ll long long</code>	<code>%lld, %llu</code>



1926

K. J. Somaiya Institute of Technology

argument types are unknown to printf

printf has to *guess* arg types from the control string.

```
#include <stdio.h>

int main() {

    int x = -1;

    printf("%d\n",x);
    printf("%u\n",x);
    printf("%hu\n",x);

    printf("%x\n",x);
    printf("%hx\n",x);

    return 0;
}
```




1926

K. J. Somaiya Institute of Technology

argument types are unknown to printf

printf has to *guess* arg types from the control string.

```
#include <stdio.h>

int main() {

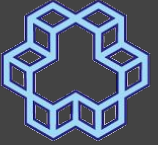
    int x = -1;

    printf("%d\n",x);
    printf("%u\n",x);
    printf("%hu\n",x);

    printf("%x\n",x);
    printf("%hx\n",x);

    return 0;
}
```

```
nasihatkon@kntu:code$ gcc streams2.c && ./a.out
-1
4294967295
65535
ffffffff
ffff
```



1926

K. J. Somaiya Institute of Technology

formatting floating points

float variables are converted to double before getting passed to printf

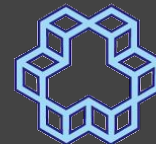
`%f` normal decimal point format

`%e, %E` scientific format

`%g, %G` Either like `%f` or `%e (%E)`

`L long double` `%Lf, %Le, %LE, %LG`

formatting floating points



1926

K. N. Toosi University of Technology

```
double x = 0.0000232345;
```

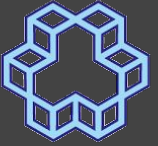
```
printf("%f\n",x);
```

```
printf("%e\n",x);
```

```
printf("%E\n",x);
```

```
printf("%g\n",x);
```

```
printf("%G\n",x);
```



1926

K. J. Somaiya Institute of Technology

formatting floating points

```
double x = 0.0000232345;
```

```
printf("%f\n",x);
```

```
printf("%e\n",x);
```

```
printf("%E\n",x);
```

```
printf("%g\n",x);
```

```
printf("%G\n",x);
```

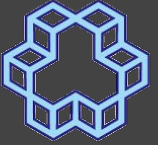
```
nasihatkon@kntu:code$ gcc streams3.c && ./a.out  
0.000023  
2.323450e-05  
2.323450E-05  
2.32345e-05  
2.32345E-05
```

strings and characters

```
char s[] = "salam";
```

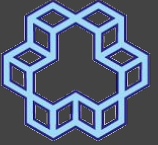
```
char c = 's';
```

```
printf("%s %c", s, c);
```



1926

K. N. Toosi University of Technology



1926

K. J. Somaiya Institute of Technology

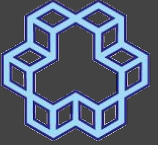
displaying pointers

Pointer size is machine-dependent

```
char s[] = "salam";  
char c = 's';  
int d = 10;  
  
char *sp = s;  
char *cp = &c;  
char *dp = &d;  
  
printf("%p %p %p\n", sp, cp, dp);
```

printing a % character

```
printf("%%30 of %f is %f \n", x, x * 0.3);
```

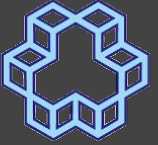


1926

K. J. Somaiya Institute of Technology

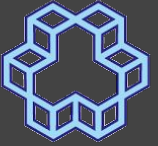
printing a % character

```
printf("%%30 of %f is %f \n", x, x * 0.3);
```



1926

K. J. Somaiya Institute of Technology



1926

K. J. Somaiya Institute of Technology

field width, left and right alignment

```
int x = 123;

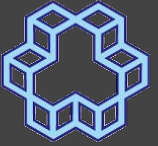
printf("%4d|\n",x);
printf("%-4d|\n",x);

printf("%6d|\n",x);
printf("%-6d|\n",x);

printf("%2d|\n",x);
printf("%-2d|\n",x);
```

```
nasihatkon@kntu:code$ gcc streams4.c && ./a.out
 123|
123 |
   123|
123  |
123|
123|
```

variable field width



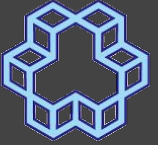
1926

K. J. Somaiya Institute of Technology

```
int x = 123;

for (int n = 4; n < 10; n++)
    printf("%*d|\n", n, x);
```

```
nasihatkon@kntu:code$ gcc streams5.c && ./a.out
123|
 123|
   123|
    123|
     123|
      123|
```



1926

K. N. T. University of Technology

other specifiers

`%+d`

`% d` print a space before positive integers

`%#x` print 0x before hex

`%#o` print 0 before octal

`%06d` print 0 before the integers

printing preceding 0's



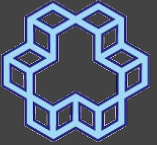
1926

K. J. Somaiya Institute of Technology

```
int x = 123;  
printf("%06d\n", x);
```

```
nasihatkon@kntu:code$ gcc streams6.c && ./a.out  
000123
```

having fun with printf!



1926

K. J. Somaiya Institute of Technology

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

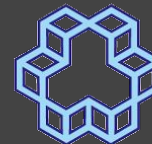
int main() {

    for (int i = 0; i < 1000; i++) {
        printf("%04d\n", i);
        sleep(1);
    }

    return 0;
}
```

```
nasihatkon@kntu:code$ gcc streams7.c && ./a.out
0000
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
```

having fun with printf!



1926

K. N. Toosi University of Technology

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

int main() {

    for (int i = 0; i < 1000; i++) {
        printf("\r%04d", i);
        fflush(stdout);
        sleep(1);
    }

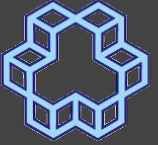
    return 0;
}
```

floating point precision

```
printf("%.3f", f);
```

```
printf("%6.3f", f);
```

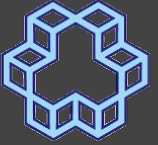
```
printf("%*.*f", m, n, f);
```



1926

K. J. Somaiya Institute of Technology

scanf



1926

K. J. Somaiya Institute of Technology

```
int x=100,y=200;  
  
scanf("%d,%d", &x, &y);  
  
printf("%d,%d\n", x, y);
```

```
nasihatkon@kntu:code$ gcc streams8.c && ./a.out  
1,2  
1,2  
nasihatkon@kntu:code$ gcc streams8.c && ./a.out  
1 2  
1,200
```


matching

```
char s[100];
```

```
scanf("%[SsCsZz]",s);
```



1926

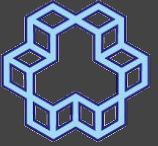
K. J. Somaiya Institute of Technology

matching

```
char s[100];
```

```
scanf("%[^.!?]",s);
```

```
int main() {  
    char s[100];  
  
    scanf("%[^.!?]", s);  
  
    printf("%s\n", s);  
  
    return 0;  
}
```



1926

K. J. Somaiya Institute of Technology

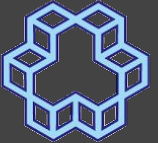
matching

```
int i,j;
```

```
scanf("%d,%d", &i, &j);
```

```
scanf("%d-%d", &i, &j);
```

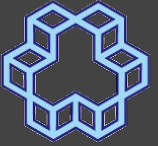
```
scanf("%d*%d", &i, &j);
```



1926

K. J. Somaiya Institute of Technology

scanning at most N characters



1926

K. J. Somaiya Institute of Technology

```
int main() {  
    char name[20];  
  
    scanf("%19s\n", s);  
  
    printf("%s\n", s);  
  
    return 0;  
}
```