



Fundamentals of Programming

session 27

Character & String processing

```
*****  
* 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 width;  
    float data[MAX_KERNEL_WIDTH];  
} ConvolutionKernel;
```

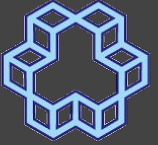
```
/* Kernels */
```

`scanf("%s"`

```
char name[20];
```

```
printf("Enter your name: ")
```

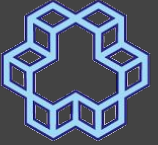
```
scanf("%s", name);
```



1926

K. J. Somaiya Institute of Technology

Remember: Strings and characters



1926

K. J. Somaiya Institute of Technology

```
#include <stdio.h>

int main() {
    char s1[] = {'S', 'a', 'l', 'a', 'm', '!', '\\n', 0};
    char s2[] = "Salam!\\n";

    printf("%s", s1);
    printf("%s", s2);

    return 0;
}
```

putchar12.c

ASCII Table

<https://commons.wikimedia.org/wiki/File%3AAscii-proper-color.svg>

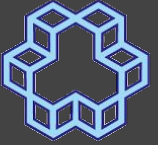


1928

K. N. Toosi University of Technology

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Print ascii code

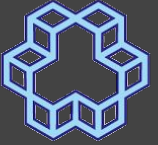


1926

K. J. Somaiya Institute of Technology

```
#include <stdio.h>

int main() {
    for (int c = 0; c < 128; c++) {
        printf("'%c' \t%d\n", c, c);
    }
}
```



1926

K. J. Somaiya Institute of Technology

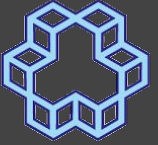
Print ascii code

```
#include <stdio.h>

int main() {
    for (int c = 0; c < 128; c++) {
        printf("'%c' \t%d\n", c, c);
    }
}
```

```
nasihatkon@kntu:code$ gcc print_ascii1.c && ./a.out
'|'      0
'|'      1
'|'      2
'|'      3
'|'      4
'|'      5
'|'      6
'|'      7
'|'      8
```

```
'7'     55
'8'     56
'9'     57
':'     58
';'     59
'<'    60
'='     61
'>'    62
'?'     63
'@'     64
'A'     65
'B'     66
'C'     67
```



1926

K. J. Somaiya Institute of Technology

Print ascii code

```
#include <stdio.h>

int main() {
    for (int c = 0; c < 128; c++) {
        printf("%c\t%d\n", c, c);
    }
}
```

```
nasihatkon@kntu:code$ gcc print_ascii1.c && ./a.out
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

handling characters in stdlib



1926

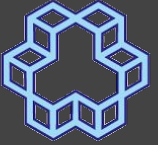
K. J. Somaiya Institute of Technology

```
behrooz@vps77786:~$ man isalpha
```

SYNOPSIS

```
#include <ctype.h>

int isalnum(int c);
int isalpha(int c);
int iscntrl(int c);
int isdigit(int c);
int isgraph(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
int isxdigit(int c);
```

1926

K. J. Somaiya Institute of Technology

handling characters in stdlib

int isalpha(int c) 

This function checks whether the passed character is alphabetic.

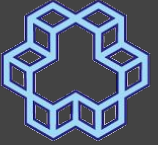
int islower(int c) 

This function checks whether the passed character is lowercase letter.

int isupper(int c) 

This function checks whether the passed character is an uppercase letter.

https://www.tutorialspoint.com/c_standard_library/ctype_h.htm



1926

K. J. Somaiya Institute of Technology

handling characters in stdlib

int isdigit(int c) 

This function checks whether the passed character is decimal digit.

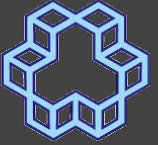
int isalnum(int c) 

This function checks whether the passed character is alphanumeric.

int isxdigit(int c) 

This function checks whether the passed character is a hexadecimal digit.

https://www.tutorialspoint.com/c_standard_library/ctype_h.htm



1926

K. J. Somaiya Institute of Technology

handling characters in stdlib

int isspace(int c) 

This function checks whether the passed character is white-space.

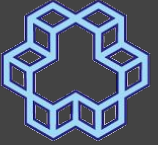
int isprint(int c) 

This function checks whether the passed character is printable.

int iscntrl(int c) 

This function checks whether the passed character is control character.

https://www.tutorialspoint.com/c_standard_library/ctype_h.htm



1926

K. J. Somaiya Institute of Technology

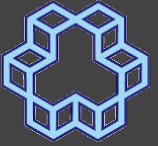
handling characters in stdlib

SYNOPSIS

```
#include <ctype.h>

int isalnum(int c);
int isalpha(int c);
int iscntrl(int c);
int isdigit(int c);
int isgraph(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
int isxdigit(int c);
```

```
int my_isdigit(int c) {
```



1926

K. J. Somaiya Institute of Technology

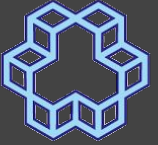
handling characters in stdlib

SYNOPSIS

```
#include <ctype.h>

int isalnum(int c);
int isalpha(int c);
int iscntrl(int c);
int isdigit(int c);
int isgraph(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
int isxdigit(int c);
```

```
int my_isdigit(int c) {
    return '0' <= c && c <= '9';
}
```



1926

K. J. Somaiya Institute of Technology

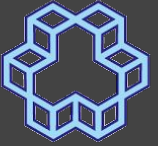
handling characters in stdlib

SYNOPSIS

```
#include <ctype.h>

int isalnum(int c);
int isalpha(int c);
int iscntrl(int c);
int isdigit(int c);
int isgraph(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
int isxdigit(int c);
```

```
int my_isalpha(int c) {
```



1926

K. N. Toosi University of Technology

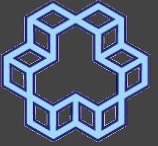
handling characters in stdlib

SYNOPSIS

```
#include <ctype.h>
```

```
int isalnum(int c);  
int isalpha(int c);  
int iscntrl(int c);  
int isdigit(int c);  
int isgraph(int c);  
int islower(int c);  
int isprint(int c);  
int ispunct(int c);  
int isspace(int c);  
int isupper(int c);  
int isxdigit(int c);
```

```
int my_isalpha(int c) {  
    return ('a' <= c && c <= 'z') || ('A' <= c && c <= 'Z');  
}
```



1926

K. J. Somaiya Institute of Technology

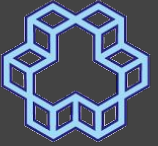
handling characters in stdlib

SYNOPSIS

```
#include <ctype.h>

int isalnum(int c);
int isalpha(int c);
int iscntrl(int c);
int isdigit(int c);
int isgraph(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
int isxdigit(int c);
```

```
int my_isspace(int c) {
```

1926

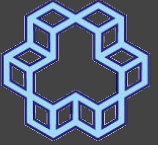
K. N. Toosi University of Technology

handling characters in stdlib

```
int my_isspace(int c) {  
    return c == ' ' || c == '\t' || c == '\r' || c == '\n' || c == '\v' || c == '\f';  
}
```

```
int isalnum(int c);  
int isalpha(int c);  
int iscntrl(int c);  
int isdigit(int c);  
int isgraph(int c);  
int islower(int c);  
int isprint(int c);  
int ispunct(int c);  
int isspace(int c);  
int isupper(int c);  
int isxdigit(int c);
```

printing ascii characters - revised



1926

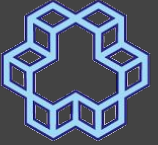
K. N. Toosi University of Technology

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

int main() {
    for (int c = 0; c < 128; c++) {

        if isprint(c)
            printf("'%c' \t%d\n", c, c);
        else
            printf("' ' \t%d\n", c);
    }
}
```

handling characters in stdlib



1926

K. J. Somaiya Institute of Technology

int tolower(int c) 

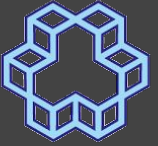
This function converts uppercase letters to lowercase.

int toupper(int c) 

This function converts lowercase letters to uppercase.

https://www.tutorialspoint.com/c_standard_library/ctype_h.htm

Example: rewrite "toupper"



1926

K. N. Toosi University of Technology

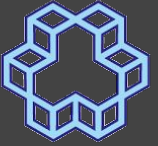
```
#include <stdio.h>

int my_toupper(int c) {

}

int main() {
    for (int c = 0; c < 128; c++) {
        printf("%c\t%c\n", c, my_toupper(c));
    }
}
```

Example: rewrite "toupper"



1926

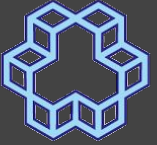
K. N. Toosi University of Technology

```
#include <stdio.h>

int my_toupper(int c) {
    return (c >= 'a' && c <= 'z') ? c + ('A' - 'a') : c;
}

int main() {
    for (int c = 0; c < 128; c++) {
        printf("%c\t%c\n", c, my_toupper(c));
    }
}
```

Converting string to integer



1926

K. J. Somaiya Institute of Technology

```
nasihatkon@kntu:code$ man atoi
```

ATOI(3)

Linux Programmer's Manual

ATOI(3)

NAME

`atoi`, `atol`, `atoll` - convert a string to an integer

SYNOPSIS

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

```
int atoi(const char *nptr);  
long atol(const char *nptr);  
long long atoll(const char *nptr);
```

Feature Test Macro Requirements for glibc (see `feature_test_macros(7)`):

DESCRIPTION

The `atoi()` function converts the initial portion of the string pointed to by `nptr` to `int`. The behavior is the same as

```
strtol(nptr, NULL, 10);
```

except that `atoi()` does not detect errors.

The `atol()` and `atoll()` functions behave the same as `atoi()`, except that they convert the initial portion of the string to their return type of `long` or `long long`.

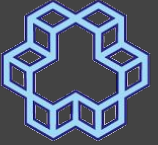
RETURN VALUE

The converted value.

Converting string to integer

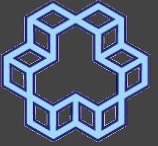
```
int i = atoi("12356");
```

```
int j = atoi("-12349");
```



1926

K. J. Somaiya Institute of Technology



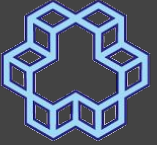
1926

K. J. Somaiya Institute of Technology

Write your own atoi

atoi1.c

```
int my_atoi(char *s) {  
    int n = 0;  
  
    while (isdigit(*s)) {  
        n = n*10 + (*s - '0');  
        s++;  
    }  
    return n;  
}
```

1926

K. N. Toosi University of Technology

Write your own atoi

atoi2.c

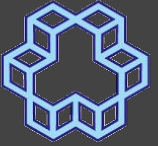
```
int my_atoi(char *s) {
    int n = 0;
    int sign = 1;

    if (*s == '-') {
        sign = -1;
        s++;
    } else if (*s == '+') {
        s++;
    }

    while (isdigit(*s)) {
        n = n*10 + (*s-'0');
        s++;
    }

    return sign * n;
}
```

Converting string to double



1926

K. N. Toosi University of Technology

ATOF(3)

Linux Programmer's Manual

ATOF(3)

NAME

`atof` - convert a string to a double

SYNOPSIS

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

```
double atof(const char *nptr);
```

DESCRIPTION

The `atof()` function converts the initial portion of the string pointed to by `nptr` to `double`. The behavior is the same as

```
strtod(nptr, NULL);
```

except that `atof()` does not detect errors.

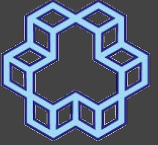
RETURN VALUE

The converted value.

Converting string to double

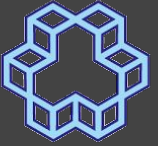
```
double x = atof(".12356");
```

```
double y = atof("-12349.000");
```



1926

K. J. Somaiya Institute of Technology



1926

K. N. Toosi University of Technology

strtol & strtod

NAME

strtol, strtoll, strtouq - convert a string to a long integer

SYNOPSIS

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

```
long int strtol(const char *nptr, char **endptr, int base);
```

```
unsigned long int strtoul(const char *nptr, char **endptr, int base);
```

NAME

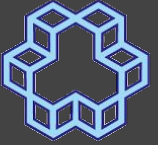
strtod, strtodf, strtold - convert ASCII string to floating-point number

SYNOPSIS

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

```
double strtod(const char *nptr, char **endptr);
```

strtol & strtod



1926

K. N. Toosi University of Technology

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

int main() {

    char msg[] = "12000 years from now!";

    char *p;

    int d = strtol(msg, &p, 10);

    printf("%d\n",d);
    printf("%s\n",p);

    return 0;
}
```

stdio

```
int putchar(int c);
```

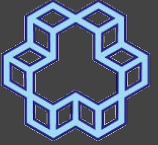
```
int getchar(void);
```

```
int puts(const char *s);
```

```
char *gets(char *s);
```

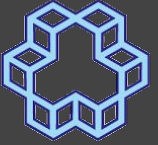
```
int sprintf(char *str, const char *format, ...);
```

```
int fscanf(FILE *stream, const char *format, ...);
```



1926

K. J. Somaiya Institute of Technology



1926

K. N. Toosi University of Technology

string manipulation (string.h)

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

```
char *strcpy(char *dest, const char *src);
```

```
char *strncpy(char *dest, const char *src, size_t n);
```

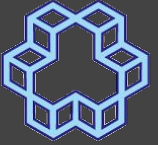
```
char *strcat(char *dest, const char *src);
```

```
char *strncat(char *dest, const char *src, size_t n);
```

```
size_t strlen(const char *s);
```

string comparison

- `s1` is equal to `s2`
- `s1` is larger than `s2`
- **Sorting names in a list**



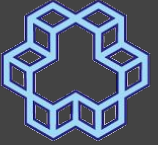
1926

K. N. Toosi University of Technology

string comparison

```
int strcmp(const char *s1, const char *s2);
```

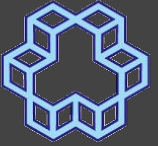
```
int strncmp(const char *s1, const char *s2, size_t n);
```



1926

K. N. Toosi University of Technology

Example

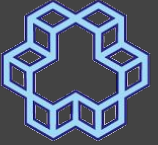


1926

K. N. Toosi University of Technology

```
int main() {  
    char s1[] = "Shir Jamshid";  
    char s2[] = "Shir Farzad";  
    char s3[] = "Shir Farhad";  
  
    printf("%d\n", strcmp(s1, s2));  
  
    return 0;  
}
```

write you own strcpy



1926

K. N. Toosi University of Technology

```
int main() {
    char s1[] = "Shir Jamshid";
    char s2[] = "Shir Farzad";
    char s3[] = "Shir Farhad";

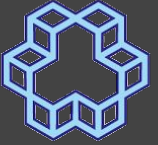
    printf("%d\n", strcmp(s1, s2));
    printf("%d\n", my_strcmp(s1, s2));

    return 0;
}
```

```
int my_strcmp(char *s1, char *s2) {
    while (*s1 == *s2 && s1 != '\0') {
        s1++;
        s2++;
    }

    return *s1 - *s2;
}
```

bubble sort



1926

K. N. Toosi University of Technology

```
void bubbleSort(char *a[], int n);
void swap(char **p, char **q);
void printStrArray(char **a, int n);

int main() {
    char *a[] = {
        "Behrooz",
        "Behrad",
        "Bahman",
        "Bahram",
        "Behnam"
    };
    int n = sizeof(a) / sizeof(a[0]);

    printStrArray(a,n);

    bubbleSort(a,n);

    printStrArray(a,n);

    return 0;
}
```

bubble sort



1926

K. N. Toosi University of Technology

```
void bubbleSort(char *a[], int n);
void swap(char **p, char **q);
void printStrArray(char **a, int n);

int main() {
    char *a[] = {
        "Behrooz",
        "Behrad",
        "Bahman",
        "Bahram",
        "Behnam"
    };
    int n = sizeof(a) / sizeof(a[0]);

    printStrArray(a,n);

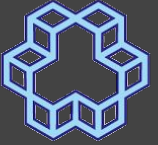
    bubbleSort(a,n);

    printStrArray(a,n);

    return 0;
}
```

```
void printStrArray(char **a, int n) {
    if (n == 0) {
        putchar('\n');
        return;
    }

    puts(a[0]);
    printStrArray(a+1, n-1);
}
```



1926

K. N. Toosi University of Technology

bubble sort

```
void bubbleSort(char *a[], int n);
void swap(char **p, char **q);
void printStrArray(char **a, int n);

int main() {
    char *a[] = {
        "Behrooz",
        "Behrad",
        "Bahman",
        "Bahram",
        "Behnam"
    };
    int n = sizeof(a) / sizeof(a[0]);

    printStrArray(a,n);

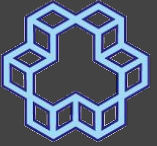
    bubbleSort(a,n);

    printStrArray(a,n);

    return 0;
}
```

```
void bubbleSort(char *a[], int n) {
    for (int m = n-1; m > 0; m--)
        for (int i = 0; i < m; i++)
            if (strcmp(a[i], a[i+1]) > 0)
                swap(&a[i], &a[i+1]);
}

void swap(char **p, char **q) {
    char *temp;
    temp = *p;
    *p = *q;
    *q = temp;
}
```



1926

K. N. Toosi University of Technology

bubble sort

```
void bubbleSort(char *a[], int n);  
void swap(char **p, char **q);  
void printStrArray(char **a, int n);
```

```
int main() {  
    char *a[] = {  
        "Behrooz",  
        "Behrad",  
        "Bahman",  
        "Bahram",  
        "Behnam"  
    };  
    int n = sizeof(a) / sizeof(a[0]);  
  
    printStrArray(a,n);  
  
    bubbleSort(a,n);  
  
    printStrArray(a,n);  
  
    return 0;  
}
```

```
void bubbleSort(char *a[], int n) {  
    for (int m = n-1; m > 0; m--)  
        for (int i = 0; i < m; i++)  
            if (strcmp(a[i], a[i+1]) > 0)  
                swap(&a[i], &a[i+1]);  
}
```

```
nasihatkon@kntu:code$ gcc strsort.c && ./a.out  
Behrooz  
Behrad  
Bahman  
Bahram  
Behnam  
  
Bahman  
Bahram  
Behnam  
Behrad  
Behrooz
```

searching strings



1926

K. N. Toosi University of Technology

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

```
char *strchr(const char *s, int c);
```

```
char *strrchr(const char *s, int c);
```

```
size_t strspn(const char *s, const char *accept);
```

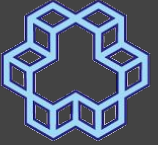
```
size_t strcspn(const char *s, const char *reject);
```

```
char *strpbrk(const char *s, const char *accept);
```


substring search

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

```
char *strstr(const char *haystack, const char *needle);
```



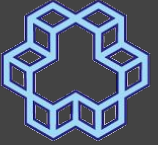
1926

K. M. Toppal University of Technology

tokenizing strings

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

```
char *strtok(char *str, const char *delim);
```



1926

K. N. Toosi University of Technology