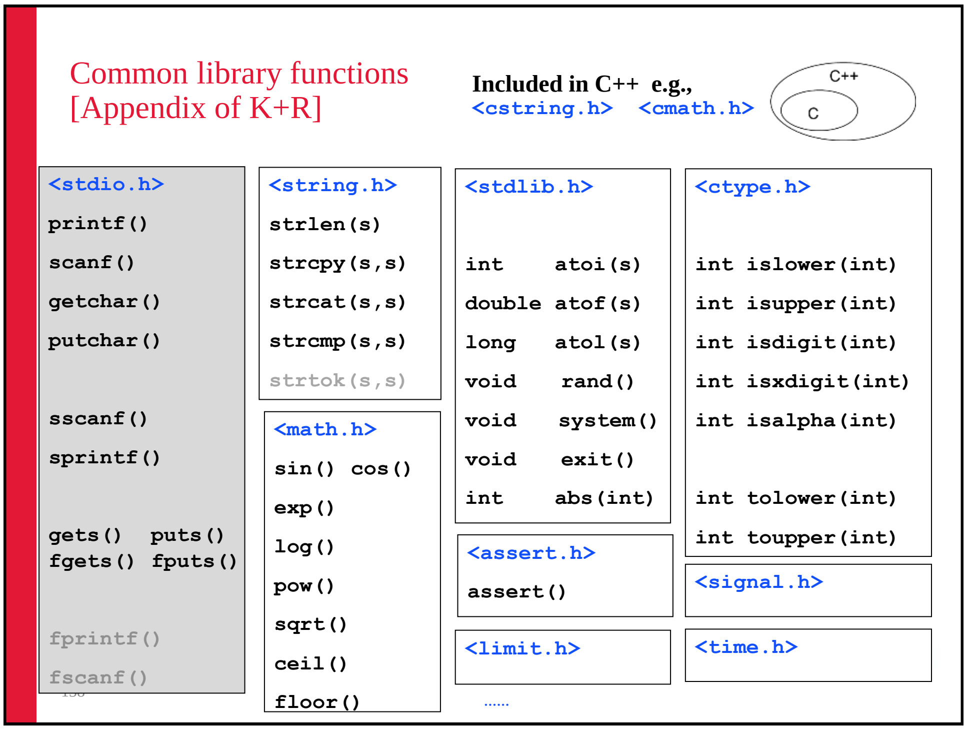


Common library functions [Appendix of K+R]

Included in C++ e.g., `<cstring.h>` `<cmath.h>`

```
<stdio.h>
printf()
scanf()
getchar()
putchar()

sscanf()
sprintf()

gets()   puts()
fgets()  fputs()

fprintf()
fscanf()
```

```
<string.h>
strlen(s)
strcpy(s,s)
strcat(s,s)
strcmp(s,s)
strtok(s,s)
```

```
<math.h>
sin()  cos()
exp()
log()
pow()
sqrt()
ceil()
floor()
```

```
<stdlib.h>

int     atoi(s)
double  atof(s)
long    atol(s)
void    rand()
void    system()
void    exit()
int     abs(int)
```

```
<assert.h>
assert()
```

```
<limit.h>
```

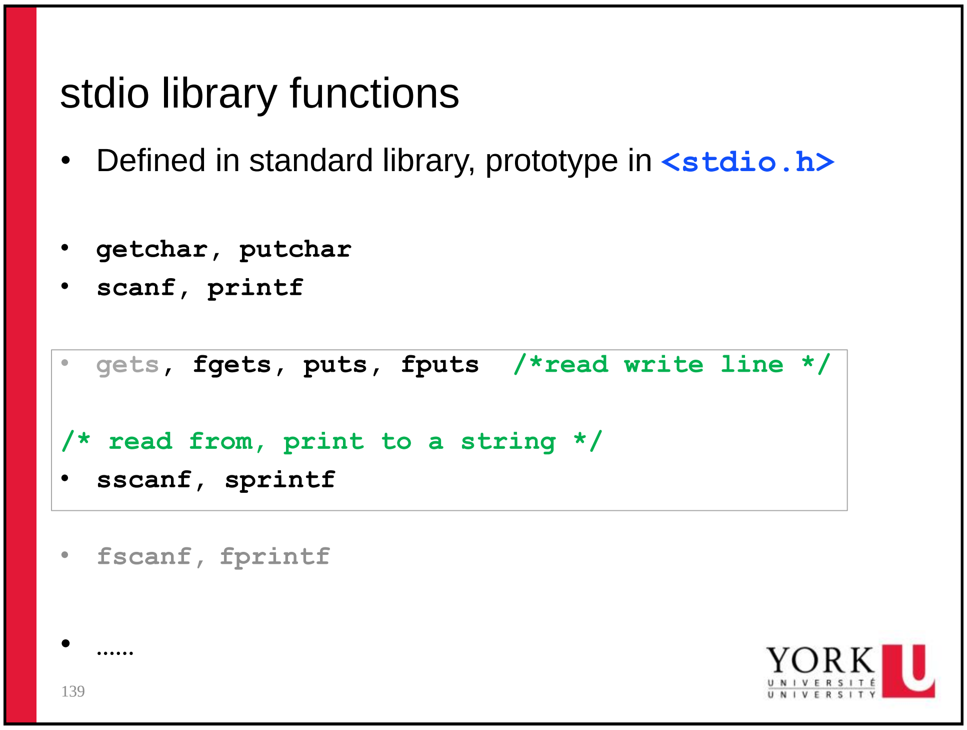
……

```
<ctype.h>

int islower(int)
int isupper(int)
int isdigit(int)
int isxdigit(int)
int isalpha(int)

int tolower(int)
int toupper(int)
```

```
<signal.h>
```

```
<time.h>
```

138



stdio library functions

- Defined in standard library, prototype in `<stdio.h>`

- **getchar, putchar**
- **scanf, printf**

- gets, **fgets, puts, fputs** `/*read write line */`

`/* read from, print to a string */`
- **sscanf, sprintf**

- fscanf, fprintf

- ……

139

YORK U UNIVERSITÉ UNIVERSITY

139

Basic I/O functions `<stdio.h>`

- **int printf (char *format, arg1, ….);**
  - Formats and prints arguments on standard output (screen or > outputFile)
  - **printf("This is a test %d \n", x)**

- **int scanf (char *format, arg1, ….);**
  - Formatted input from standard input (keyboard or < inputFile)
  - **scanf("%d %c", &x, &y)**

- **int sprintf (char * str, char *format, arg1,…..);**
  - Formats and prints arguments to char array (string) str
  - **sprintf(str, "This is a test %d \n", x)** // nothing print on stdout!

- **int sscanf (char * str, char *format, arg1, ….);**
  - Formatted input from char array (string) str
  - **sscanf(str, "%d %c", &x, &y)** // tokenize string str

140

strings: set /get in general

```
char message[20];
```

- To get from another string (literal) – `strcpy` prototype `<string.h>`
  - `strcpy(message, "hello")`
  - `str[] = "Hi";  strcpy(message , str);`   `Hi\0`

  `"john 12 2.3"   ?`

- `sprintf` -- Defined in standard library, prototype in `<stdio.h>`
  - `sprintf(message, "%s %d %f", "john",12, 2.3);`

    `"john 12 2.300000"`

    `format and then write to message`   `\0 added`
  - `sprintf(message, "%s %d-%.3f", "john",12, 2.3);`

    `"john 12-2.300"`

- `sscanf(message, "%s %d-%f", name, &age, &rate );`

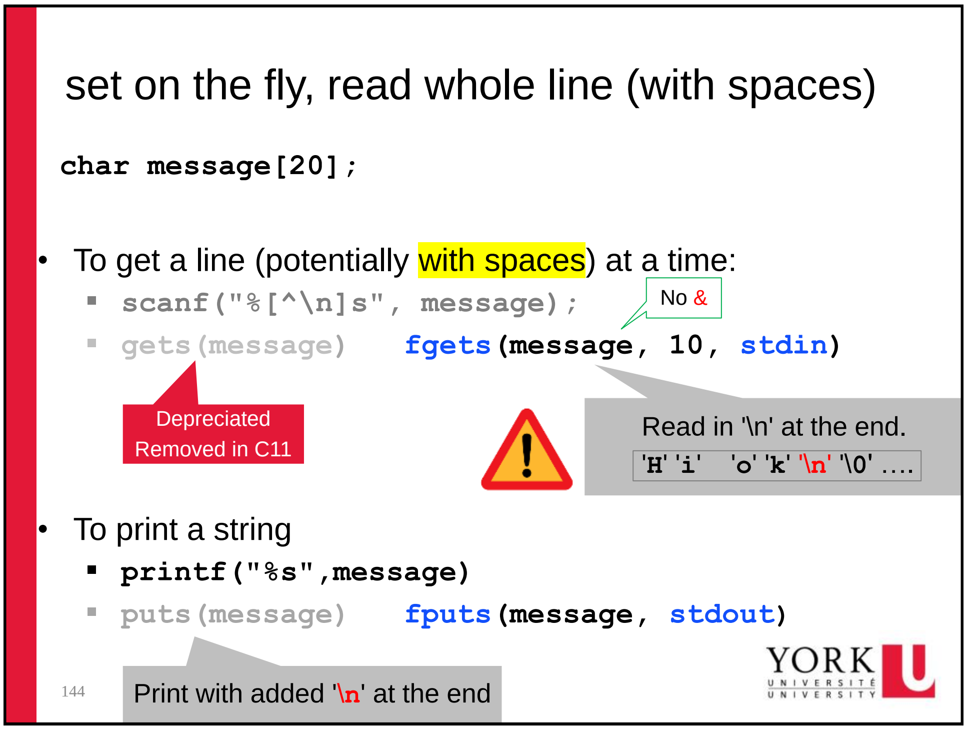
141

Way of generating/tokenizing a string   `tokenizing string message`

141

```
#include <stdio.h>

main(){
 char message [30];
 int age =20;   char name[]="john";  double rate = 4.3456;
 printf("%s %d-%f\n", name, age, rate); // john 20-4.345600
 printf("%s %d-%.3f\n", name, age, rate); // john 20-4.346

 // format and write to message
 sprintf(message, "%s %d-%.3f", name, age, rate); // no screen
 printf("%s\n", message); // john 20-4.346

 int age2; float rate2;  char name2[20];

 // tokenize message
 sscanf(message, "%s %d-%f", name2, &age2, &rate2);

 printf("%s\n", name2); // john
 printf("%d\n", age2);  // 20
 printf("%f\n",rate2);  // 4.346000
 printf("%.3f\n",rate2);  // 4.346
```

- No live lab session today and tomorrow
- Lab4 first part posted today
- SMQ1 this Friday 7pm~3am.
- Assignment1 soon



set on the fly, read whole line (with spaces)

```
char message[20];
```

- To get a line (potentially with spaces) at a time:
  - `scanf("%[^\n]s", message);`
  - `gets(message)` `fgets(message, 10, stdin)`

No &

Depreciated
Removed in C11

Read in '\n' at the end.

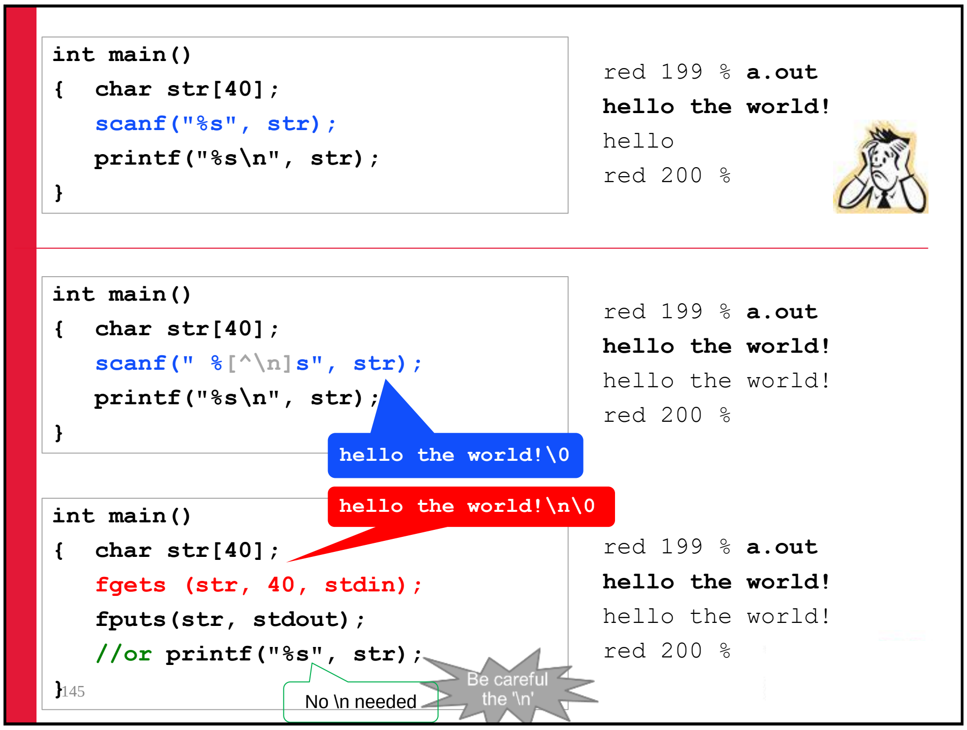
'H' 'i' 'o' 'k' '\n' '\0' ….

- To print a string
  - `printf("%s",message)`
  - `puts(message)` `fputs(message, stdout)`

Print with added '\n' at the end

144

144



```
int main()
{   char str[40];
    scanf("%s", str);
    printf("%s\n", str);
}
```

```
red 199 % a.out
hello the world!
hello
red 200 %
```

```
int main()
{   char str[40];
    scanf(" %[^\n]s", str);
    printf("%s\n", str);
}
```

hello the world!\0

```
red 199 % a.out
hello the world!
hello the world!
red 200 %
```

hello the world!\n\0

```
int main()
{   char str[40];
    fgets (str, 40, stdin);
    fputs(str, stdout);
    //or printf("%s", str);
}
```

No \n needed

Be careful the '\n'

```
red 199 % a.out
hello the world!
hello the world!
red 200 %
```

145

145

```
int main()
{   char str[40];
    fgets(str, 40, stdin);
    while (strcmp(str, "quit"))
    {
      fputs(str);   //  \n printed
      // printf("%s", str);

      // read again
      fgets(str, 40, stdin);
    }
}
```

```
red 199 % a.out
hello the world!
hello the world!
This is good
This is good
quit
quit
quit
quit
……
```

146

```
int main()
{   char str[40];
    fgets(str, 40, stdin);
    while (strcmp(str, "quit\n"))
    {
      fputs(str);   //  \n printed
      // printf("%s", str);

      // read again
      fgets(str, 40, stdin);
    }
}
```

```
red 199 % a.out
hello the world!
hello the world!
This is good
This is good
quit
red 200 %
```

```
int main()
{
    char str[40];
    while (1)                    No &
    {
       fgets (str, 40, stdin);
       if (! strcmp(str, "quit\n"))
          break;              // ==0
       fputs(str, stdout);
    }
}
```

Be careful the '\n'

147

147

```
int main()
{   char str[40];
    scanf(" %[^\n]s", str);
    while (strcmp(str, "quit"))
    {

       printf("%s\n",str);

       // read again
       scanf(" %[^\n]s", str);
     }
}
```

```
red 199 % a.out
hello the world!
hello the world!
This is good
This is good
quit
red 200 %
```

```
int main()
{
    char str[40];
    while (1)
    {
       scanf(" %[^\n]s", str);
       if (! strcmp(str, "quit"))
          break;
       puts(str);
    }
}
```

str does not contain '\n'

148

148

	Non-string (int, char, float…)	String
Read scanf, sscanf fgets…	&x	arrName
Write printf, sprintf fputs…	x	arrName

& except String

no &

```
char message[40];
char name[20];  int age, float rate, char le;

scanf("%s %d %f %c", name, &age, &rate, &le);
sscanf(message, "%s %d %f %c", name, &age, &rate, &le);

printf("%s %d %f %c", name, age, rate, le);
sprintf(message, "%s %d %f %c", name, age, rate, le);
```

149

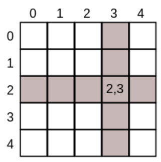
149

- Finished Ch1 – 4

- Other C materials before pointer
  - Common library functions [Appendix of K+R]
  - **2D array, table of strings**

YORK UNIVERSITÉ UNIVERSITY

150



Multi-dimension array, array of arrays

- `int arr2D[3][2];`

  // 3 rows, 2 columns

	col 0	col 1
row 0	a[0][0]	a[0][1]
row 1	a[1][0]	a[1][1]
row 2	a[2][0]	a[2][1]

- Initialization when declaring:
```
int arr2D[3][2] = {1,1,2,4,3,9};
int arr2D[3][2] = {{1,1},{2,4},{3,9}}
```

- Access: `arry2D[2][1]`   9

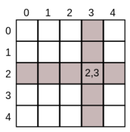
	0	1
0	1	1
1	2	4
2	3	9

- size? How stored?

Java

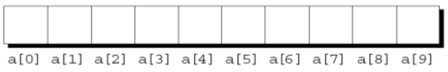
Same in Java

151



Multi-dimension array how are they stored

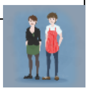
- `int arr1D[10];`

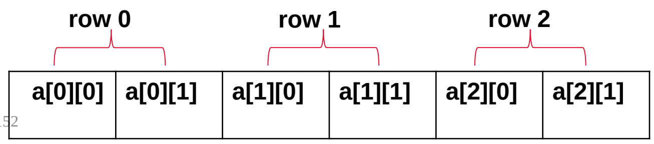


  - Size: type bytes * number of element
    - 4 * 10 = 40 bytes

- `int arr2D[3][2]`
  - Size: type bytes * row * column
    - 4 * 3 * 2 = 24 bytes

	col 0	col 1
row 0	a[0][0]	a[0][1]
row 1	a[1][0]	a[1][1]
row 2	a[2][0]	a[2][1]





152

An example

```
int main(){
   int a [3][2] = {{1,2},{3,4},{5,6}};
   printf("sizeof: %d\n", sizeof a);
   int i, j;
   for(i=0; i<3; i++){
      for(j=0; j<2; j++)
         printf("%d ", a[i][j]);
      printf("\n") ;
   }
   a[0][1] = 100;
   a[2][1] *= 10;
   printf("\n");
   for(i=0; i<3; i++){
      for(j=0; j<2; j++)
        printf("%d ", a[i][j]);
      printf("\n") ;
   }
}
```

	col 0	col 1
row 0	a[0][0]	a[0][1]
row 1	a[1][0]	a[1][1]
row 2	a[2][0]	a[2][1]

```
sizeof: 24
1 2
3 4
5 6

1 100
3 4
5 60
```

153

	col 0	col 1	col 2
row 0	a[0][0]	a[0][1]	a[0][1]
row 1	a[1][0]	a[1][1]	a[0][1]

```
int main(){
   /* 2D array declaration*/
   int arr[2][3];

   int i, j;
   for(i=0; i<2; i++) {
      for(j=0;j<3;j++) {
         printf("Enter value for arr[%d][%d]:", i, j);
         scanf("%d", &arr[i][j]);  // cell level
      }
   }
   //Displaying array elements
   for(i=0; i<2; i++) {
      for(j=0;j<3;j++) {
         printf("%d ", arr[i][j]);
         if(j==2)
            printf("\n");
      }
   }
```

```
Enter value for arr[0][0]:1
Enter value for arr[0][1]:2
Enter value for arr[0][2]:3
Enter value for arr[1][0]:4
Enter value for arr[1][1]:5
Enter value for arr[1][2]:6
1 2 3
4 5 6
```

154

Multi-dimension char array, array of strings

- Array of “strings”
  `char messages[3][6]`
  `={"Hello","Hi", "Ther"};`

	0	1	2	3	4	5
0	H	e	l	l	o	\0
1	H	i	\0	.	.	.
2	T	h	e	r	\0	.

- Size?   type bytes * row * column   1 * 3 *6 = 18 bytes

- Each row (e.g., message[0]) is a (1-D) char array (string)
  - `printf("%s", messages[0]);`   Hello
  - `printf("%d", strlen(messages[1]));`   2
  - `strcmp (messages[0], messages[2]);` a negative num
  - `printf("%c", messages[2][1]);`   h

155

Multi-dimension array, array of strings each row is a 1D string set in general

`char messages[3][10]`
Get from another string (literal)

- `strcpy` Write to the first/2nd row
  - `strcpy(messages[0], "hello")`
  - `str[] = "Hi";  strcpy(messages[1], str);`

- `sprintf sscanf`
  - `sprintf(messages[1], "%s %d %f", "john",12,2.3)`

    format and then write to 2nd row
  - `sscanf(messages[2],"%s %d %f", name, &age,&wage`

    tokenizing the 3rd row

157

157

Multi-dimension array, array of strings each row is a 1D string set on fly

- To read a word into a row
  `scanf("%s", messages[1]);`
  // read into row 2

	0	1	2	3	4	5
0	H	e	l	l	o	\0
1	H	i	\0	.	.	.
2	T	h	e	r	\0	.

- To read in a line with space into a row at a time:
  - `scanf("%[^\n]s", messages[0]);` No &
  - `gets(messages[0])` `fgets(messages[0], 10, stdin`

    depreciated

    read into the first row

    append \n at end
- To print a row
  - `printf ("%s", messages[0]);`
  - `puts(messages[1])` `fputs(messages[2], stdout)`

158

158

```
={ {'J','a','v','a','\0'},
   {'P','y','t','h','o','n','\0'},
   {'C','+','+','\0'},
   {'H','T','M','L','\0'},
   {'S','Q','L','\0'} };
```

```
char lang[5][10]={"Java", "Python", "C++", "HTML","SQL"};
for(int i=0;i<5;i++)
  printf("%s %d", lang[i], strlen(lang[i]));;

lang[0] = "Kotlin";

// we can do char level,
lang[0][0]='K'; lang[0][1]='o' ……

// or, use lib function copy the String
strcpy(lang[0], "Kotlin"); // valid

printf("Enter 5 rows");
for(i=0; i<5; i++)
  scanf("%s",lang[i]); or fgets(lang[i], 10, stdin);

for(int i=0;i<5;i++)
  fputs(lang[i],stdout);
```

J	a	v	a	\0					
P	y	t	h	o	n	\0			
C	+	+	\0						
H	T	M	L	\0					
S	Q	L	\0						

? How scanf(), strcpy (), sscanf(), fgets(), change argument if pass-by-value?

? How could Mr. Main's paper get color changed?