I/O Functions

Preliminary Assumption

- ASCII Code & value for different characters
- C Function structure: mainly return value and argument list
- Data type in C
- Variables in C
- Statements in C
- Operators in C
- Expression in C

Input/Output Functions (Standard)

- **integer getchar():** Get a single character
- **integer putchar(integer):** Print a single character
- **character address gets(character address variable):** Get a full line
- **integer puts(string):** Print a full line
- **integer scanf(control, arguments):** Takes input based on control
- **integer printf(control, arguments):** Shows output based on control
- …

integer getchar()

- Get a **single** character
- Return **ASCII** value of input
- See the example source code

```
#include<stdio.h>

int main(){
    char character;
    int integer;

    character = getchar();
    integer = getchar();

    return 0;
}
```

integer putchar(integer)

- Prints a **single** character
- Argument as an **ASCII** value
- Return **EOF** if error
- See the example source code

```
#include<stdio.h>

int main(){
    char character;
    int integer;

    character = getchar();
    integer = getchar();

    putchar(character);
    putchar(integer);

    return 0;
}
```

More about integer putchar(integer)

```
##include<stdio.h>

int main(){

    putchar('c');
    putchar(115);
    putchar('a'+4);

    return 0;
}
```

Character is given as **regular** form

Character is given as **ASCII** value

Character is given as ASCII **expression**

character address gets(character address variable)

- Gets a group of characters
- Takes input upto newline ‘\n’ character
- Replace the ‘\n’ with NULL in C
- Return an address of the starting character of the input
- See the example source code

```
#include<stdio.h>

int main(){

    char line[100];

    gets(line);

    return 0;
}
```

Forget about it now

character address gets(character address variable)

- Gets of charac
- Takes inpu
- Retu
- See th

```
stdio.h>

e);

    return 0;
}
```

Deprecated !!!

Forget abou w

Alternative for gets():
character address fgets(character Array, integer, file stream)

- Needs to mention maximum length of input
- Need to mention input stream
- Rest of the things same as gets()
- See the example source code

```
#include<stdio.h>

int main(){

    char line[100];

    fgets(line, 100, stdin);

    return 0;
}
```

integer puts(string)

- Print the given string
- Return EOF when error occured
- See the example source code

```
#include<stdio.h>

int main(){

    puts("Hail CSE");

    return 0;
}
```

Conversion Characters in Format Specifier

Symbol	Meaning	Symbol	Meaning
d	integer	c	character
f	float	lf	double
s	string	lld, I64d	long long int
...	...	...	...

Escape Characters

Symbol	Meaning
\n	New line
\t	Tab
\”	A double quotation
\\	A back slash
\b	Backspace
...	...

integer printf(control, arguments)

- Multi-purpose printing function
- Prints anything based on control string
- See the example source code

```
#include<stdio.h>

int main(){

    printf("Sentence #1");
    printf("Sentence #2");
    printf("Sentence #3");
    printf("Sentence #4");

    return 0;
}
```

Output:

```
Sentence #1Sentence #2Sentence #3Sentence #4
```

integer printf(control, arguments)

- Multi-purpose printing function
- Prints anything based on control string
- ‘\n’ is used for new line
- See the example source code

```
#include<stdio.h>

int main(){

    printf("Sentence #1\n");
    printf("Sentence #2");
    printf("Sentence \n#3");
    printf("Sentence #4");

    return 0;
}
```

```
Output:
Sentence #1
Sentence #2Sentence
#3Sentence #4
```

integer printf(control, arguments)

- Multi-purpose printing function
- Prints anything based on control string
- ‘\n’ is used for new line
- Use of other escape characters
- Use of variables
- See the example source code

```
#include<stdio.h>

int main(){
    int intVariable = 10;
    double doubleVariable = 3.14;

    printf("The integer value = %d\n",intVariable);
    printf("The double value = %lf\n",doubleVariable);

    return 0;
}
```

```
Output:
The integer value = 10
The double value = 3.140000
```

integer printf(control, arguments)

- Multi-purpose printing function
- Prints anything based on control string
- ‘\n’ is used for new line
- Use of other escape characters
- Use of variables
- Use of expressions
- See the example source code

```
#include<stdio.h>

int main(){
    int intVariable = 10;
    double doubleVariable = 3.14;

    printf("The integer value = %d\n",intVariable);
    printf("The integer value = %d\n",120);
    printf("The double value = %lf\n",doubleVariable);
    printf("The double value = %lf %d\n",doubleVariable, 10);
    printf("The double value = %d\n", 12*25/7+100%13);

    return 0;
}
```

```
Output:
The integer value = 10
The integer value = 120
The double value = 3.140000
The double value = 3.140000 10
The double value = 51
```

integer printf(control, arguments)

- Multi-purpose printing function
- Prints anything based on control string
- ‘\n’ is used for new line
- Use of other escape characters
- Use of variables
- Use of expressions
- Print in fixed length
- …
- See the example source code

```
#include<stdio.h>

int main(){
    int intVariable = 10;
    double doubleVariable = 3.14;

    printf("The integer value = %6d\n",intVariable);
    printf("The double value = %10.2lf\n",doubleVariable);
    printf("The integer value = %06d\n",intVariable);
    printf("The double value = %010.2lf\n",doubleVariable)

    return 0;
}
```

```
Output:
The integer value =     10
The double value =       3.14
The integer value = 000010
The double value = 0000003.14
```

integer scanf(control, arguments)

- Multi-purpose input function
- Needs Container or variable
- Gets anything based on control string
- See the example source code

```
#include<stdio.h>

int main(){
    char varc;
    int vari;
    double vard;

    scanf("%c%d%lf", &varc, &vari, &vard);

    return 0;
}
```

```
Input:
A 100 3.7
```

```
Variable State:
varc = ‘A’
vari = 100
vard = 3.7
```

integer scanf(control, arguments)

- Multi-purpose input function
- Needs Container or variable
- Gets anything based on control string
- Taking string without whitespaces
- See the example source code

```
#include<stdio.h>

int main(){
    char varc[100];

    scanf("%s", varc);

    return 0;
}
```

Forget about it now

```
Input:
hello
```

```
Variable State:
varc = "hello"
```

integer scanf(control, arguments)

- Multi-purpose input function
- Needs Container or variable
- Gets anything based on control string
- Taking string without whitespaces
- Taking string with whitespaces
- See the example source code

```
#include<stdio.h>

int main(){
    char varc[100];

    scanf("%[^\n]", varc);

    return 0;
}
```

Forget about it now

```
Input:
Hello cse duian
```

```
Variable State:
varc = “Hello cse duian”
```

integer scanf(control, arguments)

- Multi-purpose input function
- Needs Container or variable
- Gets anything based on control string
- Taking string without whitespaces
- Taking string with whitespaces
- More about square-bracket
- Formatted input
- …
- See the example source code

```
#include<stdio.h>

int main(){
    int vari;

    scanf("%3d", &vari);

    return 0;
}
```

Takes at most 3 digit integer

```
Input:
123456789
```

```
Variable State:
vari = 123
```

Practice

- Write a C program which takes 3 integer as input and prints the summation of the 3 values.
- Write a C program which takes a line (length not more than 30) from user and prints the line as right aligned in a 30 character long line.
- Write a C program which takes the initial velocity and the final velocity with the occupied time of a car and prints the acceleration of the car.
- Write a C program which takes necessary input and print the result of the given expression below:

$$\left(b \times a^n + \tfrac{d}{c}\right)\%N;\ c \neq 0$$

Thank You