<C programming>

{By Zakir Hossain};

By Mahim Arif(1DM)

Scanned with CamScanner

নাম	-	পৃষ্ঠা

* অধ্যায় (১)- (১-৬)

* অধ্যায় (২) - (৭- ৯)

* অধ্যায় (৩) - (১০ - ১১)

* অধ্যায় (৪) - (১২-

Scanned with CamScanner

অধ্যায় ২
মৌলিক ধারণা

* C Prog: এর extension (.C).

* প্রথম C Prog:

=> কমান্ডলাইন: Printf, if, else, for ইত্যাদিকে machine language এ পরিণত করার জন্য কমান্ডলাইন ব্যবহার করা হয়।

=> IDE or Intergraded Development Environment:

```
#include <stdio.h>
int main()
{
    Printf ("Hello World!\n");
    return 0;
}
```

→ মানে কোনো কিছু মুক্ত করা।

→ .h মানে এটা একটা Header file.

→ standard input output.

মেইন ফাংশন

এটি একটি ফাংশন, Printf মানে হলো Print formatted. (এটি একটি লাইব্রেরী ফাংশন).

→ এর মানে হলো Main ফাংশনে শূন্য return করা।

```
Pritf(" Hi");
return 0;
```

→ এগুলোকে বলে statement, প্রতি statement এর শেষে (;) সেমি-কোলন দিতে হয়।

*[(; না দিলে compilation পারেনা হয়).]

* Data type এবং Variable:

* ভ্যারিয়েবল হচ্ছে একটি নাম যা দিয়ে computer memory তে Data রাখা হয়।

* ভ্যারিয়েবল এর মধ্যে যা রাখবে (যেমন 1, 2, 3..., a, b, c, X, Y, Z) এগুলো হচ্ছে Data-Type. Numeric ও হতে পারে, character ও হতে পারে।

Data type:

- int data type
- Char " "
- float " "
- double " "

} প্রধান 4টি Data type. এছাড়াও আছে void, যাকে Value less বলা যেতে পারে।

Scanned with CamScanner

⊛ int data type:

→ integer quantity (অবিভাজ্য সংখ্যা, ১, ২, ৩...)

→ size ~~2bit/16~~ 2 Byte or 16 Bit (1 byte = 8 bit).

→ Range (−32768 → +32767)

=> int distance = 40;

distance variable টা ডিক্লার করলাম।

ডিক্লার করার সাথে সাথে তার একটি মান নিলাম, যাকে বলে value-Assign করা।

* int value print করার জন্য printf("%d \n"); দিতে হয়,

→ %d কে বলে place holder.

→ Every data-type has different ph

→ %d means display integer.

=> Data type এবং তাদের Placeholder: এবং size

Data type	Placeholder	size
int	%d	2bytes
char	%c	1 byte
float	%f	4 bytes
double	%lf	8 bytes

Scanned with CamScanner

⊛ Character data type:

- → single char (a, b, c, N, z ....) বুঝায়,
- → Size 1 byte (8 bit).
- → Range (−128 → +127)
- → ভ্যারিয়েবল ডিক্লার করার নিয়ম:

```
#include<stdio.h>
int main()
{
char ch = 'A';
printf("%c\n", ch);
return0;
}
```

variable.

ch = 'A'; ← ভ্যারিয়েবলের মধ্যে ~~char~~ ক্যারেক্টার Assign করলে single কোটেশনের মধ্যে 'A' লিখতে হবে,

⊛ float data type:

- → floating point Number (দশমিক সংখ্যা ১০.৫, ৭.৮ ..)
- → দশমিকের পর ৬ (six) ঘর পর্যন্ত নির্ভুল ভাবে ন্যায়ে সংরক্ষণ করতে পারে,

⊛ double data type:

- → ৬ ঘরের বেশি সংখ্যা দশমিকের পর নির্ভুল ভাবে নিতে হলে,
- → 8 byte (64 bit) size.
- → দশমিকের পর 15 ঘর পর্যন্ত নির্ভুল ভাব মান সংরক্ষণ করে,

Scanned with CamScanner

⊛ Modified data type:

↳ int, double, float, char - কে সংশোধন করে আরো চারটি Data type তৈরি করা যায়।

- signed
- unsigned
- long
- short

⇒ Int - এর অন্যান্য প্রকার:

[16 bit = $(2^{16}-1)$]

- signed int (+, - বলে দিতে হয়).
- unsigned int (+, - বলে দেওয়া হয় না).
- long int
- short int
- long long int

⇒ Constant বা ধ্রুবক:

- integer constant → যেকোনো integer number (1, 2, 15, 20 ...)
- floating point constant → n দশমিক সংখ্যা. (5.2, 9.872) ...
- character constant → single কোটেশনের মধ্যে লিখা হয়। একটি single const character ('A', 'X', 'g') ...
  → শুধু একটি character থাকে।
- string constant
  যখন কোটেশনের ভেতর এক বা একাধিক character লিখলে তা আমরা string বলি। ("Hello") ...
  constant.
  ("5.629")
  যখন কোটেশনে দেয়
  (Not Num anymore) Num থাকলেও তা string হয়ে যায়।

Scanned with CamScanner

Type	Bit size	Range	Type	Bit size	Range
char	8	-127 থেকে 127	signed short int	16	-32767 থেকে 32767
unsigned char	8	0 থেকে 255	long int	32	-2147483647 থেকে 2147483647
signed char	8	-127 থেকে 127	long long int	64	$-(2^{63}-1)$ থেকে $(2^{63}-1)$
int	16/8	-32767 থেকে 32767	signed long int	32	-2147483647 থেকে 2147483647
unsigned int	16/8	0 থেকে 65535	unsigned long int	32	0 থেকে 4294967295
signed int	16/8	-32767 থেকে 32767	unsigned long long int	64	0 থেকে $(2^{64}-1)$
short int	16	-32767 থেকে 32767	float	32	1.2E-38 → 3.4E+38
unsigned short int	16	0 থেকে 65535	double	64	2.3E-308 → 1.7E+308
			long double	80	3.4E-4932 → 1.1E+4932

*** Escape sequence:** (নতুন লাইন।) (\n).

Escape seq:	কাজ
\a	Alert (beep, Bell), added in c89
\b	Backspace
\f	Formfeed
\n	Newline (Line Feed)
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\\	Backslash
\'	Single quotation mark.
\"	Double quotation mark

Scanned with CamScanner

⊛ কীভাবে (C prog:) ASCII মান বের করব?

A → ASCII মান?

```
# include <stdio.h>
int main ()
{
    char ch = 'A';
    printf ("%d", ch);
    return 0;
}.
```

নরমাল (%c) এর বদল (%d) দিলে ASCII মান output দিবে।

⊛ 65 → নরমাল মান?
ASCII

```
char ch = 65;
printf ("%c", ch);
```

[(← ' ' quotatio দিবনা)]

Scanned with CamScanner

: অধ্যায়-২য় :
: অপারেটর :

⊛ অপারেন্ড: অপারেটর হলো যে ভ্যারিয়েবল বা যে ডেটার উপর কাজ করে তাকে বলে।

⊛ Assignment operator: ভ্যারিয়েবলে কোন Value Assign করার কাজ।

↳ How to use : Identifier = expression

ভ্যারিয়েবল — ভ্যালু

যেমন:

Pi = 3.1416; / Pi = 22/7;

Assigment operators:
= (Equal to)
+= (Plus equal to)
-= (minus equal to)
*= (product equal to)
/= (Division equal to)
%= (Mode equal to)

⊛ (AO - Assigment op:)

⊛ Equal to Assignment operator:

যেমন:
1) x = 5; [এখানে 5, x এর মধ্যে Assign করা হলো]
2) Y = 10;
3) Pi = 3.1416;
4) Z = x+y+Pi;

[একইভাবে.]

⊛ += (plus equ: to) AO:→

Expressio 1 += Expression 2;

↳ এর মানে (Exp 1 = Exp 1 + Exp 2).

ব্যাখ্যা: মনে করি,
x = 5; যদি লিখি x += Y তাহলে, x এর মান হবে x = (x+Y) অর্থাৎ 8।
Y = 3;

⊛ -= (Minus equ to):→

X = 5;
Y = 3;

[X -= Y] => [X = X - Y] => [X = 3]

Scanned with CamScanner

একই ভাবে :

$* = [x *= Y] \Rightarrow [x = x * Y]$.

$/ = [x /= Y] \Rightarrow [x = x / Y]$

$\% = [x \%= Y] \Rightarrow [x = x \% Y]$

⊛ গাণিতিক অপারেটর:

গাণিতিক গণিতে আমরা যে সকল operators ব্যবহার করেছি তাই হলো Arithmetic operators, যেমন, $(+, -, \times, \div, \%)$, ইত্যাদি।

MODE OP:

(%) একে Mode/reminder বলে, এটি মূলত ভাগশেষ value

Mode এর মান Always positive,

Mode " " " পূর্ণ সংখ্যা হবে, যেমন ১৫.৫%৪ => ৩.৫, কিন্তু এর মান হবে ৩,

⊛ Unary (একনারী) operators:

↳ c prog: এ যে সকল operator একটি চলকের উপর কাজ করে নতুন মান দেয়,

↳ Most used unary operators: (—) (Minus sign)

↳ যেমন : $x = -3$,

এখানে, $+3, -3$ value টি pstv, ngtv বুঝায়, কিছু না দিলে Normally plus ধরে নেয়,

↳ এখানে দুটি Unary op:

- → Increment op: (++) { যে varible এর উপর বসে তার মান ১ বাড়িয়ে দেয়, $x = 5$ হলে, $x++$ / $++x$ মান হবে 6
- ↳ Decrement op: (--) { opposite of (++).

⊛ Logical operators:

Relational operators

৪ প্রকার

1) '<' Less than
2) '<=' Less then or equal to
3) '>' Greater then
4) '>=' Greater then or equal to

ধরি,

X = 5;
Y = 6;

যদি, (X < Y) লিখি এবং এ সত্য (5<6) তাই ~~output~~ এর মান হবে 1, যদি expression টি মিথ্যা (X > Y) হতো তাহলে মান হতো 0।

Equality operators

1) '==' Equal to
2) '!=' Not eq. to

X = 5;
Y = 6;

এর মান হলো X, Y এর মান সমান.

∴ (X == Y) টি ভুল বা মিথ্যা, তাই মান আসবে 0।

∴ (X != Y) হয় তবে মান হবে 1, কারণ এ সত্য।

Logical operators

1) '&&' And
2) '||' or.

X = 5;
Y = 6;
Z = 10;

(X < Y) && (Z > Y).

এখানে বেয়ে পারে সত্য হতে হয়। তাহলে মান আসবে 1, আর যদি যে কোন এক পাশে সত্য হয় এবং মান 1 হয় তখন or op: ব্যবহার করে।

যেমন:

(X < Y) || (X > Z)

এখানে, শুধু বামের টা সত্য তাই 1 হয়ে যাবে, যদি দুইটি মিথ্যা হয় তাহলে মিথ্যা।

⊛ Conditional op:

condition ? Exp 2 : Exp 3

সত্য হলে, নির্বাচিত হবে, মিথ্যা হলে এটি নির্বাচিত হবে

```
i = 5;
Z = (i<8)? Hi : Bye
```

যদি i এর মান 8 এর চেয়ে ছোট হয় তাহলে Print করবে Hi, যদি না হয় তাহলে " " Bye।

Scanned with CamScanner

অধ্যায়: ৩
ইনপুট & আউটপুট

getchar & putchar

⊛ **getchar:** এটি একটি লাইব্রেরী ফাংশন। এর দ্বারা single character কম্পিউটারে input নেওয়া হয়।

1| character variable = getchar();

Prg: এর মাঝে নিচের মত করে লিখা হয়:

1| char x; ⟶ মানে বুঝানো হলো এটি একটি character type variable

2| x = getchar(); ⟶ যা মান লিখবে তা।

⊛ **putchar:** এটি দ্বারা single character console-এ দেখানোর জন্য।

1| putchar (character variable);

Prg: এর মাঝে এভাবে:

1| char x = 'x'; ⟶ বুঝানো হয় যে, এটি একটি character data type variable.

2| putchar (x); ⟶ 'x' এর মান আউটপুট ডিভাইসে দেখাবে।

⇒ বড় হাতের অক্ষর output দিতে:

```
char x;
x = getchar();
putchar (touppper (x));
```

(বড় হাতের দিলে বড় হাতের output দিবে)

⇒ ছোট " " " " :

```
char x;
x = getchar();
putchar (tolower (x));
```

⟶ (" " ")....

Scanned with CamScanner

⊛ scanf & printf :

Data input

- → character
- → Number
- → string.

Data output

⊛ putchar দিয়ে একটি মাত্র character কম্পিউটারে দেখানো যেত, কিন্তু printf দিয়ে একাধিক।

Control string	Meaning
%c	Single character in/output এর জন্য ব্যবহৃত হয়
%d	signed decimal integer
%e	e নোটেশনে signed Floating point এর জন্য
%f	signed Floating point এর জন্য
%g	সাধারণ নোটেশনে signed Floating. Point এর জন্য
%i	signed decimal int এর জন্য।
%o	Unsigned octal int. বিএর জন্য
%s	String data item output দিবে.
%u	Unsigned decimal int এর জন্য
%x	Unsigned Hexadecimal data এর জন্য

Scanned with CamScanner

অধ্যায় : ৪

Control statement

* if

=> if (expression) statemen

↑ যদি এটি true হয় তাহলে if এর ভেতর ঢুকবে আর False হলে ঢুকবে না.

=> একটি if এর ভেতর আরেকটি if ব্যবহার করলে তাকে Nested if বলা হয়।

* if - else

if (expression) statement 1.
else (") statement 2.

↳ if এর মান False হলে else এর ভেতর যা আছে তা print করবে। [else এ কোন condition দেওয়া যায় না। else-if এ condition দেওয়া যায়).

* while (loop)

=> While (condition) statement.

↳ একটা cond. দেওয়া হয়, যতক্ষণ পর্যন্ত তা সত্য থাকে ততক্ষণ while লুপটি চলবে।

Scanned with CamScanner

* do-while (loop)

do statement while (expression);

↳ কিছু একটা কর, যতক্ষণ পর্যন্ত condition সত্য হয়.

```
int num = 0;
do {
    printf ("%d\n", num);
    num++;
}while (num <=9);
return 0;
}
```

output

0
1
2
3
4
5
6
7
8
9

↳ এভাবেও 3, করা যাবে (short cut).
২য় Bracket ছাড়া.

↳ do {
statement 1;
statement 2;
statement 3;
- - - - - - -
} while (expression);

যখন, only একটা statement নিয়ে কাজ করব,

```
do
printf ("-%d\n", num++);
while (num <=9);
```

Scanned with CamScanner

* for (loop)

✓ তিন অংশ রয়েছে।

```
for(exp1; Exp2; exp3) statement.
```

* exp1: * ১ম অংশ * প্রাথমিক মান দেওয়া হয় * Initial অংশ
* Exp2: * ২য় অংশ * শর্ত দেওয়া হয়। * logical Exp. (T/F)
* exp3: * প্রাথমিক মানটিকে পরিবর্তন মাধ্যমে বড়। * তৃতীয় অংশ

```
int i;
for (i=0; i<=10; i++){
    printf("%d\n",i);
}
```

Scanned with CamScanner