# **পয়েন্টার (Pointer)**

আমরা যারা Computer Science নিয়ে পড়ালেখা করছি , প্রথম দিকে দু’টি বিষয় আমাদের কাছে অনেক দুরুহ মনে হয়।

১। Pointer

২। Recursion

Recursion ভীতি ব্যাপারটা আসলে একটু স্বাভাবিক । বলা হয়ে থাকে, পৃথিবীর অধিকাংশ নতুন প্রোগ্রামারদের মধ্যে একটি বিষয়ে মিল খুজে পাওয়া যায়, সেটি হল “I hate recursion”। তবে দু’টো topic-ই দারুন গুরুত্বপূর্ন । আমি চেষ্টা করব, এই দুটো টপিক নিয়ে, একজন শিক্ষার্থীর দৃষ্টিকোন থেকে আলোচনা করার ।

অনুরোধ করব আস্তে আস্তে সময় নিয়ে পড়ার। NO তাড়াহুড়া।

OK…… So, Pointer নিয়ে শুরু হয়ে যাক ।

আমি আশা করছি , যেহেতু Pointer বুঝতে আগ্রহী, তার মানে , Loping, Array সম্পর্কে তুমি জান বা ধারণা আছে ।

Pointer এর শুরুতেই (\*) <--- Asterisk, চিহ্ন এর সাথে পরিচিত হয়ে নাও। Pointer related সকল প্রোগ্রামেই আমরা এটি ব্যবহার করব ।

**Pointer Variable এর কাজ, দেখতে কেমনঃ**

আমরা যখন প্রোগ্রামে কোন Variable (সাধারণ ভাবে যা করি) ক্রিয়েট করি , কম্পিউটার এর memory-তে সেটি একটি যায়গা দখল করে নেয় । For Example:-

int coc; // datatype and variable declaration

এখানে coc Integer Variable-টি কম্পিউটারের memory-তে কিছু যায়গা দখল করল। Integer Variable সাধারণত 2 bytes/৪ bytes ( Compiler ভেদে ) যায়গা দখল করে নেয় । যদি আমি নাম দেই, ধরি certain Memory Location-এ Integer Variable-টি (coc) আছে।

(Try reading slowly from here)

এখন যদি আমি চাই,-- ঐ Variable (coc) এর--- Memory Location আমি User-কে Show করব । তবে সে ক্ষেত্রে Variable-টির Memory Address store করার জন্য আমার একটি Variable প্রয়োজন হবে । সাধারণ , Integer Variable দিয়ে এই Address show করা যায় না।

এর জন্য একটি ‘special variable’ নির্মাতারা নির্ধারণ করে দিয়েছেন। তা হল, Pointer Variable। এই Variable গুলোর সাহায্যে , আমরা কোন সাধারণ Variable এর Memory Address store করতে পারি এবং পরবর্তীতে আমরা এই Address ব্যবহার করে বিভিন্ন Process করতে পারি। যা আমরা ধীরে ধীরে দেখব।

Pointer Variable তৈরী করতে ভেরিয়েবল নাম এর আগে (\*) চিহ্ন ব্যবহার করতে হয়।

For Example:-

int \*rr3;

এখানে , rr3 একটি Integer Type Pointer Variable, যা অন্য একটি সাধারণ Integer Variable এর Memory Address save করবে।

**Pointer Operator:**

Pointer Variable নিয়ে Process এর জন্য দু’টি Operator রয়েছে ।

১. Address Operator 🡪 &

২. Indirection Operator 🡪 \*

**Address Operator:**

Address Operator প্রকাশ করা হয় “&” দ্ধারা ।এই Operator সম্পর্কে আশা করি তোমরা অনেকেই পরিচিত। scanf function দ্ধারা ভেরিয়েবল এ value assign করার সময়, এই Address Operator (যেটাকে **Ampersand** বলা হয়) এর সাহায্যে value input নিয়েছ ।

সরাসরি উদাহরণ দিয়ে বুঝাই:

#include<stdio.h>

int main()

{

int p, \*pv ;

------(code)

-------(code)

-------(code)

return 0;

}

এখানে আমি একটি Integer Variable (p) এবং একটি Integer Pointer variable (\*pv) তৈরী করলাম।

Pointer Variable-টি একটি Address store করবে, (যদি আমি পরবর্তীতে প্রোগ্রামে চাই) আর Integer Variable-টি একটি Integer Number store করবে।

কোন Variable memory-এর কোন location-এ আছে, তা জানা যায় এই Address Operator (&) ব্যবহার করে।

Address Operator ব্যবহার করে “p” Variable এর Location জানা যাবে, (&p)🡨 expression এর সাহায্যে ।

আর location-টি আমরা save করব , “pv” Pointer Variable এ।

Let’s write the program:

1. Create a variable name p

2. Create a pointer variable name pv

3. Assign address of p to the (pv) Pointer Variable. (which means, pv will point to the variable p)

#include<stdio.h>

int main()

{

int p,\*pv;

pv= &p; //Now pv point or hold the address of p variable . not \*pv = &p

}

**এখানে একটি ছোট confusion এর কথা বলি**:

Once you have *created* the pointer variable by using (\*pv), to use it in the program, you don’t have to use (\*) again, if u do, it will mean something else, which we will learn soon.

**Indirection Operator:**

OK, Indirection Operator পড়ার পর, অনেক গুলো confusion তৈরী হতে পারে। আমি আশা করছি, সময় নিয়ে পড়বে ।

প্রথমত , আমি বলেছি , কোন Pointer Variable তৈরী করার সময়, (\*) [asterisk] চিহ্ন ব্যবহার করতে হবে। একবার তৈরী হয়ে গেলে, Pointer Variable ব্যবহারে এটি (\*) বারবার দেওয়ার দরকার নেই।

উপরের example-এ , আমরা p নামে একটি ভেরিয়েবল ব্যবহার করেছি , যার অ্যাড্রেস ছিল \*pv Pointer Variable এর কাছে (pv= &p) ।

যদি p ভেরিয়েবলটির কোন value থাকত, ধরে নেই , (p = 6),

then------- (Pay attention !),

**“The data Item represented by p, can be accessed by the expression (\*pv), where** \* **is a unary operator, called the *indirection operator*”.**

উপরের ইংলিশটি এখন একটু বোঝার চেষ্টা করি:

“The data Item represented by p” – মানে p ভেরিয়েবল যে Data Item (value) represent করছে, তার মান ৬ । (এখানে variable এর data item বলতে ৬ কে বোঝানো হয়েছে) ।

“--- can be accessed by the expression \*pv”—বলছে p ভেরিয়েবল এর ডাটা আইটেম এ এক্সেস করা যাবে (\*pv ) দ্ধারা । ( একইভাবে কিন্তু pointer variable তৈরী করা হয়েছে। দু’টো ভিন্ন জিনিস remember) ।

**SO…Quick Recap**:-

1. p is an integer variable and p=6;

2 .pv is a pointer variable (remember the rules of creating a pointer variable)

3. pv holds the address of p integer variable by using address operator (pv=&p).

4. Now, (\*pv) represents the value of the p variable (it is possible because of point no 3,as pv holds the address of p variable)

**কয়েকটি ছোট প্রোগ্রামের সাহয্যে , উপরের ৪টি পয়েন্ট ক্রিস্টাল ক্লিয়ার হবে বলে আমি মনে করি।**

**একটি CODE লিখার চেষ্টা কর নিজে—স্টেপ গুলো আমি বলছি ।**

1. Create an integer variable name q and assign a value.

2. Create an integer pointer variable name mo;

3. Assign the memory address of q to the mo pointer variable

4. Now, print the value of q variable, without using “q”. (Use indirection operator)

নিজে চেষ্টা কর, না পারলে নিচের CODE-টি দেখ :

#include<stdio.h>

int main()

{

int q=6; //step 1

int \*mo; //step 2

mo=&q; //step 3

printf("\nvalue of q variable is %d\n",\*mo); //step 4

return 0;

}

Output আসবে 6

আমরা চাইলে , indirectly q variable এর মান পরিবর্তন ও করে দিতে পারি ।

উপরের code-এ যদি আমরা q variable এর value 6 এর পরিবর্তে 9 করে দেই,তবে অবশ্যই Pointer (address operator and indirection operator) এর সাহায্যে ।

নিজে একটু চিন্তা কর, স্টেপ গুলো দেখ:

1. Create an integer variable name r and assign a value 6.

2. Create an integer pointer variable name bn;

3. Assign the memory address of r to the bn pointer variable

4. Now change the value of the r variable using indirection operator

5. Now, print the value of r variable.

**না পারলে, নিচের CODE-টি দেখ এবং বুঝে নাও:**

#include<stdio.h>

int main()

{

int r=6; //step 1

int \*bn; //step 2

bn=&r; //step 3

\*bn=9; //step 4 and that is the changing step

printf("\nvalue of p variable is %d\n",p); //step 5

return 0;

}

প্রথমে আমি , r variable এর value দিয়েছি 6, পরে pointer variable bn ব্যবহার করে এর value 9 করেছি এবং পরবর্তীতে print করেছি।

Pointer এর বেসিক ধারণা এতটুকু । Pointer এর এই Operator গুলো এবং তাদের কাজ সম্পর্কে ধারণা থাকলে, pointer এর Code বুঝা অনেকটাই সহজ হয়ে যায় । আশা করি, pointer এর কোড নিয়ে পড়তে বসতে আর জড়তা কাজ করবেনা।

Pointer Notation এর সাহায্যে Array data Sort ও করা যায়।

আর Data Structure এর বিভিন্ন topic, যেমন, Linked List, TREE , Binary TREE ইত্যাদি বুঝতে হলে অবশ্যই পয়েন্টার সম্পর্কে জানতে হবে।  
Recursion নিয়ে ও আলাদাভাবে লিখার ইচ্ছা আছে ।  
 **আমার অনিচ্ছাকৃত ভুলগুলো জানাতে ভুল করবে না ।**  
  
 সবার জন্য শুভকামনা।

**HAPPY CODING**.

Mohammad Imran Hossain

09.10.2015

CSE (2nd Year)

IIUC

Contact: [fb/mohammadimran.hossain.35](https://www.facebook.com/mohammadimran.hossain.35)