

The total number of sectors:

$$120*200 + 120*240 + 120*320 = 91200$$

so, the total disk capacity:  $91200 * 4096 = 373555200$

bytes. = 356 MB. (Ans)

To get the transfer rate, first, pick the cylinder zone with the largest number of sectors (320). This is the cylinder zone that will yield the greatest transfer rate. It's probably near the outside of the disk (since there's more surface area to work with).

Now, multiply the number of sectors by the number of bytes per sector ( $320 * 4096$ ) and you get 1310720. This is the number of bytes you can retrieve per disk rotation.

The number of rotations per second is 100/60, approximately 1.67. Multiply that by 1310720 bytes per rotation, and you get 2188902, or about 2.08 megabytes per second. (Ans)

#### Structure Programming

**[Syllabus: BPSC CS:]** Computer Programming: Introduction to computer programming. Assembling language programming. Problem solving techniques, algorithm specification and development. Programming style, testing and debugging. Program design techniques: Structured and modular program design. Programming languages and paradigms: classification.

**Programming in C:** Data type, statements, control structures, arrays, pointers, strings, functions, preprocessor directives, structures, unions and bit-fields, files.

**NTRCA CS:** Overview of C, C++ and JAVA; Constants, variables and data types; Operator & Expression; Managing Input & Output Operators; Decision making and branching; Decision making and looping. Arrays; Handling of character string; User-defined function; Structure and union; Pointers; File management.

**NTRCA ICT(BM):** সি প্রোগ্রামিং ল্যাঙুেজেজ ডাটা টাইপ, এবং সংক্ষিপ্ত প্রোগ্রাম এবং ফোর্মাট।

**NTRCA ICT:** Fundamentals of C programming; Introducing C's Program Control Statements; Data types, Variables and Expressions; Exploring Arrays and Strings; Understanding Pointers and Functions; Console and File I/O; Structures and Unions.]

## Structure Programming (C Program)

### প্রশ্ন ১. প্রোগামিং ভাষা কী? [what is programming language]

উত্তরঃ কম্পিউটারকে আমাদের প্রয়োজনীয় নির্দেশাবলী জানানোর জন্য এক বিশেষ ধরনের ভাষা ব্যবহার করা হয়। এ বিশেষ ধরনের ভাষা কম্পিউটার প্রোগ্রাম রচনার জন্য ব্যবহৃত হয়ে থাকে বলে একে প্রোগামিং ভাষা বা Programming Language বলে।

### প্রশ্ন ২. কম্পিউটার প্রোগ্রাম কী? [what is computer program]

[সহজরী মেইল ইঞ্জিন-২০১০, NTRA-2014, 15]  
উত্তরঃ কম্পিউটারের সাহায্যে কোনো একটি বিশেষ কার্য সম্পাদন বা সমস্যা সমাধানের জন্য রচিত বা লিখিত ধারাবাহিক কতভালো বিশেষ নির্দেশাবলী (Instruction) বা কমান্ডকে কম্পিউটার প্রোগ্রাম বলে।

### প্রশ্ন ৩. Machine Language কাকে বলে? [what is machine language]

উত্তরঃ যে ভাষায় সমস্যা সমাধানের জন্য বাইনারি সংখ্যা (0,1) ব্যবহার করে নির্দেশ সহ সজিয়ে প্রোগ্রাম রচনা করা হয়, তাকে Machine Language বলে।

### প্রশ্ন ৪. মধ্য ভরের ভাষা কী? [what is mid level programming language]

উত্তরঃ ১৯৬০ সালের দিকে ইঁরেজী ভাষাকে আধাৰ্য নিয়ে কম্পিউটারের প্রোগ্রাম রচনার কুরার জন্যে যে ভাষা অবিহৃত হয় তা মধ্য ভরের ভাষা নামে পরিচিত।

### প্রশ্ন ৫. উচ্চ ভরের ভাষা কী? [what is the high level programming language]

[NTRCA-2014]  
উত্তরঃ উচ্চ ভরের ভাষা হলো ইঁরেজী ভাষা এবং আরো বেশি কাঠামোবৰ্ক। উচ্চ ভরের ভাষা আমেরিকান ন্যাশনাল স্ট্যার্ড ইঙ্গিটিউটের নির্দেশ মেনে বেশির ভাগ উচ্চ ভরের ভাষা তৈরী হয়। উচ্চ ভরের ভাষা সহজে লেখা যায়, সংকলন করা যায় এবং ভুল সংশোধন করা যায়। উচ্চ ভরের ভাষার বৈশিষ্ট্য হল ইহা বালিজাক ও বৈজ্ঞানিক প্রয়োগের ভাষা। এটা বহু প্রযোগ ও চৰ্তৰ্ব অজন্য ভাষা। তাই উচ্চ ভরের ভাষাকে বহু প্রয়োগের ভাষা বলা হয়। searching, sorting এবং process এর ক্ষেত্ৰে বড় ধরনের সুবিধা লাভ করা যায়।

### প্রশ্ন ৬. প্রোগ্রাম তৈরীর ধাপসমূহ কী? [NTRCA-2015, 17]

উত্তরঃ কম্পিউটারের সাহায্যে বিভিন্ন সমস্যা সমাধানের জন্যে কতকগুলো পদক্ষেপ নিতে হয়, এ পদক্ষেপ সমূহকে প্রোগ্রাম তৈরীর ধাপ বলে।  
ধাপসমূহ নিম্নলিপি:  

- সমস্যা চিহ্নিত করা।
- প্রোগ্রামের input/output নির্নয় করা।
- প্রোগ্রামের Algorithm develop করা।
- Flowchart তৈরী করা।
- Programming ভাষা নির্ধারণ করা।
- Program রচনা করা।
- compile করা।
- documentation
- রক্ষণাবেক্ষণ করা।

### প্রশ্ন ৭. Assembly Language/Low level language কাকে বলে? [BPDB-2004]

উত্তরঃ যে ভাষায় সমস্যা সমাধানের জন্য বিশেষ ধরনের সংক্ষিপ্ত সংকেত বা নেমেনিক (ADD, SUB, MUL, DIV, LOAD, INC

ইত্যাদি) ব্যবহার করে প্রোগ্রাম রচনা করা হয়, তাকে Assembly Language বলে।

### প্রশ্ন ৮. কয়েকটি উচ্চভরের ভাষার নাম লেখ।

উত্তরঃ BASIC, COBOL, C, C++, PASCAL, FORTRAN, JAVA, C#, PYTHON, PROLOG, RUBI

### প্রশ্ন ৯. IDE এর প্রক্রিয়া কী?

উত্তরঃ Integrated Development Environment.

### প্রশ্ন ১০. Source প্রোগ্রাম কাকে বলে?

উত্তরঃ উচ্চভরের এবং আসেবলি ভাষায় রচিত প্রোগ্রামকে Source প্রোগ্রাম বলে।

### প্রশ্ন ১১. অনুবাদক প্রোগ্রাম কাকে বলে?

উত্তরঃ যে প্রোগ্রাম কম্পিউটারের Source প্রোগ্রাম (যে ভাষায় প্রোগ্রামটি লেখা হয়) কে machine ভাষায় অনুবাদ করে সে প্রোগ্রামকে অনুবাদক প্রোগ্রাম বলে।

### প্রশ্ন ১২. অনুবাদক প্রোগ্রাম কোথায় প্রকার?

উত্তরঃ তিনি প্রকার। যথাঃ

ক. আসেবলার (Assembler)

ব. কম্পাইলার (Compiler)

গ. ইন্টারপ্রেটার (Interpreter)

### প্রশ্ন ১৩. আসেবলার কী?

উত্তরঃ ইহা আসেবলি ভাষায় লিখিত প্রোগ্রামকে মেশিন ভাষায় অনুবাদ করে। এটি কোন কাজের সংক্ষিপ্ত শব্দ দ্বারা প্রোগ্রাম রচনা করা হয়।

### প্রশ্ন ১৪. আসেবলারের কাজ লিখ।

উত্তরঃ আসেবলারের উচ্চতৃপূর্ণ কাজগুলো দেওয়া হলোঃ

a. নেমেনিক কোডকে মেশিন ভাষায় অনুবাদ করে।

b. প্রত্যেক নির্দেশ টিক আছে কিনা পরীক্ষা করে, ঠিক না থাকলে ঠিক করে।

c. সব নির্দেশ ও ডাটা প্রধান মেরিটে রাখে।

### প্রশ্ন ১৫. কম্পাইলার কী? [BPSC-2014, NTRCA-2011]

উত্তরঃ কম্পাইলার উচ্চভরের ভাষার Source প্রোগ্রামকে Machine প্রোগ্রামে অনুবাদ করে। কম্পাইলার সম্পূর্ণ প্রোগ্রামটিকে এক সঙ্গে পড়ে এবং এক সঙ্গে অনুবাদ করে। তিনি তিনি উচ্চভরের ভাষার জন্য ভিন্ন কম্পাইলার লাগে। কোনো নির্দিষ্ট কম্পাইলার একটি মাত্র উচ্চভরের ভাষাকে মেশিন ভাষায় অনুবাদ করতে পারে। যেমন যে কম্পাইলার BASIC কে মেশিন ভাষায় অনুবাদ করতে পারে তা FORTRAN কে মেশিন ভাষায় অনুবাদ করতে পারে না।

### প্রশ্ন ১৬. কম্পাইলারের কাজ লিখ।

উত্তরঃ কম্পাইলারের উচ্চতৃপূর্ণ কাজগুলো দেওয়া হলো -

a. Source প্রোগ্রামকে Machine প্রোগ্রামে অনুবাদ করা।

b. প্রোগ্রামকে লিঙ্ক করা।

c. প্রোগ্রামে কোনো ভুল থাকলে তা জানানো।

d. ঘোজনে Source প্রোগ্রামকে প্রিন্ট করে।

### প্রশ্ন ১৭. ইন্টারপ্রেটার কী? [NTRCA-2011]

উত্তরঃ ইন্টারপ্রেটার এক ধরনের প্রোগ্রাম যাহা উচ্চভরের ভাষাকে এক লাইন করে পড়ে ও মেশিনের ভাষায় রূপান্তর করে।

## Structure Programming (C Program)

### প্রশ্ন ১৮. ইন্টারপ্রেটারের কাজ লিখ।

উত্তরঃ নিম্নে ইন্টারপ্রেটারের উচ্চতৃপূর্ণ কাজগুলো দেওয়া হলোঃ

(ক) উচ্চভরের ভাষাকে মেশিনের ভাষায় রূপান্তর করা।

(খ) ইহা এক লাইন পড়ে ও অনুবাদ করে।

(গ) ইহা প্রতিটি লাইনের ভুল প্রদর্শন করে অনুবাদ কাজ বন্ধ করে দেয়।

(ঘ) ডিবাগিং ও টেস্টিংয়ের ক্ষেত্ৰে দ্রুত কাজ করে।

### প্রশ্ন ১৯. আলগরিদিম কাকে বলে?

উত্তরঃ কোন সমস্যা সমাধানের ধাপসমূহকে ভাষাগতভাবে লিপিবদ্ধ করাকে আলগরিদিম বলে।

### প্রশ্ন ২০. আলগরিদিমের বৈশিষ্ট্য লিখ।

উত্তরঃ নিম্নে আলগরিদিমের বৈশিষ্ট্য দেওয়া হলোঃ

(ক) সহজবোধ্য হতে হবে।

(খ) কাজের উদ্দেশ্য সূচিটি হতে হবে।

(গ) প্রত্যেকটি ধাপ স্পষ্ট হবে যাতে যেকোন প্রোগ্রামের সহজে বুঝতে পারে।

(ঘ) প্রোগ্রাম পরিবর্তন ও পরিবর্ধনে সহজয়তা করবে।

(ঙ) প্রোগ্রামের ভুল নির্ণয় করা সম্ভব হবে।

### প্রশ্ন ২১. ফ্লোচার্ট কী?

উত্তরঃ ফ্লোচার্ট হচ্ছে এক ধরণের রেখাচিত্র যার সাহায্যে একটি আলগরিদিমের ধাপ সমূহকে চিত্রের মাধ্যমে প্রকাশ করে।

### প্রশ্ন ২২. ফ্লোচার্ট কোথায় প্রকার?

উত্তরঃ ২ প্রকার। ক. সিস্টেম ফ্লোচার্ট খ. প্রোগ্রাম ফ্লোচার্ট।

### প্রশ্ন ২৩. সিস্টেম ফ্লোচার্ট কাকে বলে?

উত্তরঃ কোন সংগঠনের সকল কাজ একটি চিত্রের মাধ্যমে প্রকাশ করাকে সিস্টেম ফ্লোচার্ট বলে।

### প্রশ্ন ২৪. প্রোগ্রাম ফ্লোচার্ট কী?

উত্তরঃ কোন প্রোগ্রামের আলগরিদিম রেখাচিত্রের সাহায্যে প্রকাশ করাকে প্রোগ্রাম ফ্লোচার্ট বলে।

### প্রশ্ন ২৫. ডিবাগিং কাকে বলে?

উত্তরঃ প্রোগ্রাম ভুল ত্বরিত করাকে ডিবাগিং বলে।

### প্রশ্ন ২৬. Documentation (ডকুমেন্টেশন) কাকে বলে?

উত্তরঃ ভুল সংশোধনের পর প্রোগ্রাম টিকমতো কাজ করলে তা ভবিষ্যতে ব্যবহারের জন্য সং

## Structure Programming (C Program)

প্রশ্ন ৩৮. কম্পাইলার ও ইন্টারপ্রেটারে মধ্যে পার্থক্য কি ? [BPSC-2014, Immigration & Passport-2014, NTRCA-2014]

উত্তর: পার্থক্য নিচে দেওয়া হল:

কম্পাইলার	ইন্টারপ্রেটার
১. কম্পাইলার একটি অনুবাদক প্রোগ্রাম যা উচ্চভাবের প্রোগ্রাম ভাষার একটি সম্পূর্ণ প্রোগ্রামকে একসাথে অনুবাদ করে।	১. ইন্টারপ্রেটারও একটি অনুবাদক প্রোগ্রাম যা উচ্চভাবের প্রোগ্রাম ভাষাকে লাইন বাই লাইন অনুবাদ করে।
২. ফলে প্রোগ্রাম নির্বাচিত জন্য সহজে প্রয়োজন।	২. ফলে প্রোগ্রাম নির্বাচিত জন্য বেশি সহজ প্রয়োজন।
৩. কম্পাইলার দ্বারা একটি প্রোগ্রাম একবার অনুবাদ করা হলে প্রতিবার কাজের পূর্বে পুনরায় অনুবাদ করার প্রয়োজন হয় না।	৩. ইন্টারপ্রেটার দ্বারা একটি প্রোগ্রাম একবার অনুবাদ করা হলে প্রতিবার কাজের পূর্বে পুনরায় অনুবাদ করার প্রয়োজন হয় না।
৪. কম্পাইলার দ্বারা একটি প্রোগ্রাম অনুবাদ করলে পূর্ণাঙ্গ প্রোগ্রাম রূপান্তরিত হয়।	৪. ইন্টারপ্রেটার দ্বারা একটি প্রোগ্রাম অনুবাদ করলে পূর্ণাঙ্গ প্রোগ্রাম রূপান্তরিত হয় না।
৫. ডিবলিং এবং টেস্টিং এর ক্ষেত্রে স্বতন্ত্র সম্পর্ক।	৫. ডিবলিং এবং টেস্টিং এর ক্ষেত্রে স্বতন্ত্র সম্পর্ক।

### ব) সি প্রোগ্রামিং :

■ সি একটি প্রোগ্রামিং ভাষা। সি নির্মাণ করেন ডেনিস রিচি, বেল শ্যার্ডে ৭০ এর দশকে। ভাষাটি তৈরির প্রথম উদ্দেশ্য ছিল ইন্টিনিয়ার অপারেটিং সিস্টেম-এর কোড সোর্ক করা যাবার জন্য, কিন্তু অভিযোগেই এটি একটি বহু ব্যবহৃত ভাষার পরিণত হয়।

-In 1988, the American National Standards Institute (ANSI) has formalized the C language.

- C has been written in assembly language.

(ক) সি এর ব্যবহার :

- Database Systems
- Language Interpreters
- Compilers and Assemblers
- Operating Systems
- Network Drivers
- Word Processors

(খ) C-এর জনপ্রিয়তার কারণ :

- অন্ধকার High Level প্রোগ্রামিং ল্যাঙ্গুেজের মধ্যে সর্ব সুবিধা প্রিম্যান্ড যায়।
- Best প্রোগ্রামিং ল্যাঙ্গুেজ যা খুব স্বতন্ত্র প্রিম্যান্ড যায়।
- C language নির্ভরযোগ্য এবং ব্যবহার করা সহজ।
- C language হল Structured Language
- আধুনিক প্রোগ্রামিং এর ধরণ সি এর উপর ভিত্তি করে।
- এটি বিভিন্ন কম্পিউটারের প্র্যাটফর্মে কম্পাইল করা যায়।
- বিশ্ববিদ্যালয়সমূহ তাদের Courseware এর মধ্যে C Programming কে Preferred করে।

গ্রন্থ ১. সি দিয়ে তৈরি করা হয়েছে এরকম কিছু প্রোগ্রাম এর তালিকা লিখ।

উত্তর: মজার বিষয় হল আপনি যেই Windows টি ব্যবহার করছেন সেটি ও প্রোগ্রাম সি দিয়ে তৈরি করা হয়েছে। নিচে আরও কিছু তালিকা দেওয়া হল।

- Operating Systems(উইন্ডোজ, ডেস্ক্যাপ্টর)
- Language Compilers(কম্পাইলার)
- Assemblers(অ্যাসেবলারস)
- Text Editors(টেক্সট এডিটর)
- Print Spoolers(প্রিন্ট স্পুলার, যা দিয়ে প্রিন্টিং এর কাজ করা হয়)
- Network Drivers(নেটওর্ক ড্রাইভার)
- Modern Programs(আধুনিক প্রোগ্রাম)
- DataBase(ডাটাবেস)
- Language Interpreters
- Utilities (বিভিন্ন প্রয়োজনীয় প্রোগ্রাম ও অ্যাপ্লিকেশন)

প্রশ্ন ২. সি এর কয়েকটি জনপ্রিয় কম্পাইলার নাম লিখ।

- উত্তর:
- (code::blocks)
  - টার্বো সি (Turbo c)
  - বোরল্যান্ড (Borland)
  - মাইক্রোসফট ভিন্যুল সি++ ( Microsoft visual c++ )
  - গ্রেলি সি (gcc)
  - অ্যাপ্লেস এক্স কোড (Apple xcode) ইত্যাদি।

প্রশ্ন ৩. সি এর অধ্যান বৈশিষ্ট্য কোনো কি কি ?

উত্তর:

- (ক) সহজে শিখা যায়।
- (খ) স্ট্রাকচার্ট ভাষা।
- (গ) সি দ্বারা কার্যকর প্রোগ্রাম তৈরি করা যায়।
- (ঘ) এটি সো লেভেল ল্যাঙ্গুেজের এর সাথে কাজ করতে পারে।
- (ঙ) সব ধরনের প্র্যাটফর্মে এটি দিয়ে কাজ করা যায়।
- (ঁ) এই ল্যাঙ্গুেজে যে যত ভাল করবে, সে অন্য যেকোনো ল্যাঙ্গুেজে তার চেয়েও অনেক ভাল করবে।

প্রশ্ন ৪. C program এর বেসিক গঠন সোর্ক কি ?

উত্তর:

Documentation Section
//প্রোগ্রাম সম্পর্কে সম্বল দেখা হয় এই অংশে
link section
//সিস্টেম লাইব্রেরি ফাংশনে সম্বল প্রোগ্রামের লিঙ্ক করা হয় এই অংশে
Definition Section
// সকল symbolic constants define করা হয়।
Global declaration Section
// global ডেনিয়েবল এবং user define ফাংশন ঘোষণা করা হয়।

গ্রন্থ ২. সি দিয়ে তৈরি করা হয়েছে এরকম কিছু প্রোগ্রাম এর তালিকা লিখ।

## Structure Programming (C Program)

### Main() function

```
{
    Declaration Section
    Executable Part
}

// এভিটি প্রোগ্রামে main () ফাংশন থাকতে হবে। প্রোগ্রাম main function থেকে execute তর হয়।
```

### Sub-Program Section

Function 1
Function 2
.....
Function N

// সকল user define ফাংশন ঘোষণা করা হয়।

■ **Pseudo Code**: Pseudo Code হচ্ছে একটি প্রোগ্রাম বা একটা অ্যালগরিদমের ধাপ তথ্য সাধারণ মানদণ্ডের উপযোগী করে লেখা কিছু কোড। এভালো কোন প্রোগ্রাম ল্যাঙ্গুেজে ব্যবহার উপযোগী করে দেবা হয় না। এমন ভাবে দেবা হয় যেন মানুষ বুকতে পারে।

### Flowchart In Programming [NTRCA-2015]

Symbol	Purpose	Description
Flow line	Program এর Logic প্রবাহ নির্দেশ করে	
Terminal (Stop/Start)	Used to represent start and end of flowchart.	
Input /Output	Input and output operation এর জন্য ব্যবহার করা হয়।	
Processing	Arithmetic operations and data-manipulations এর জন্য ব্যবহার করা হয়।	
Desicion	একাধিক শর্তের উপর ভিত্তি করে সিদ্ধান্ত দেয়ার জন্য ব্যবহার করা হয়।	
Predefined Process/Function		

### ALGORITHM & FLOWCHART

1. Algorithm & Flowchart to convert temperature from Fahrenheit to Celsius.

Iterative Algorithm:

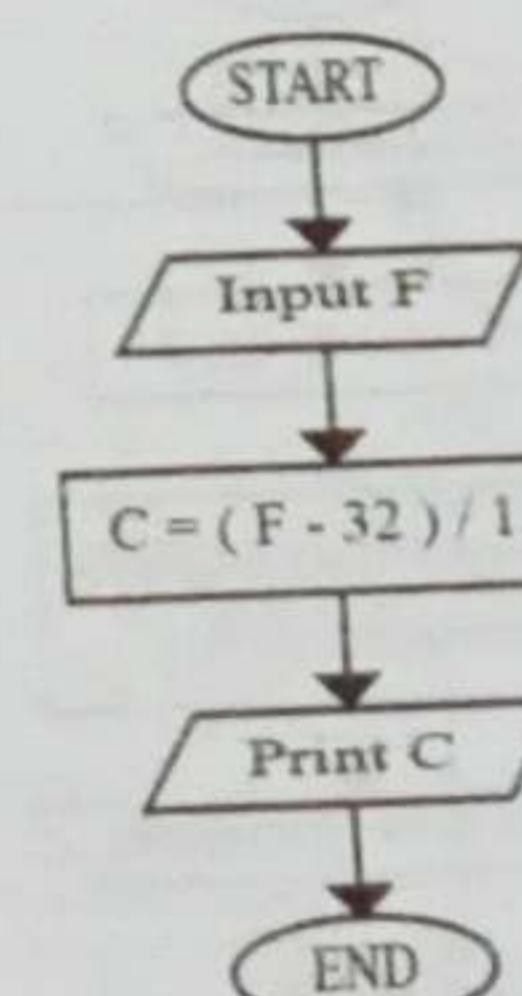
Step-1: Start

Step-2: Input temperature in Fahrenheit say F

Step-3: C = 5.0/9.0 (F - 32)

Step-4: Display Temperature in Celsius C

Step-5: Stop



■ প্রোগ্রাম সি এ ইনপুট / আউটপুট প্রোগ্রাম সি ল্যাঙ্গুেজের ইনপুট এবং আউটপুট এর জন্য ANSI স্ট্যান্ডার্ড দ্বারা অনেক লাইব্রেরি ফাংশনে নির্দিষ্ট করা হয়েছে। প্রোগ্রাম সি তে সবচেয়ে বেশি ব্যবহৃত হয় ফাংশন printf() এবং ফাংশন scanf() যথাক্রমে ইনপুট নেওয়ার জন্য এবং আউটপুট দেওয়ার জন্য।

উদাহরণ ব্রহ্ম:

Sample code of C "Hello World" program:

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

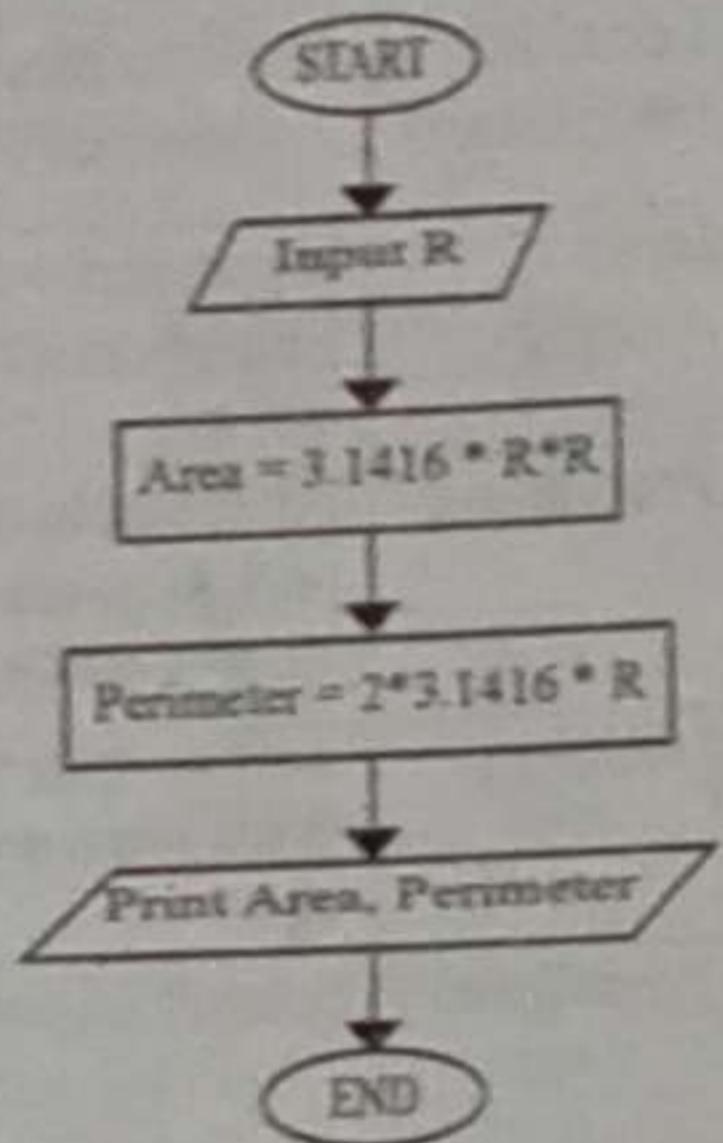
Output : Hello, World!

## Structure Programming (C Program)

### 2. Algorithm & Flowchart to find Area and Perimeter of Circle

#### Iterative Algorithm:

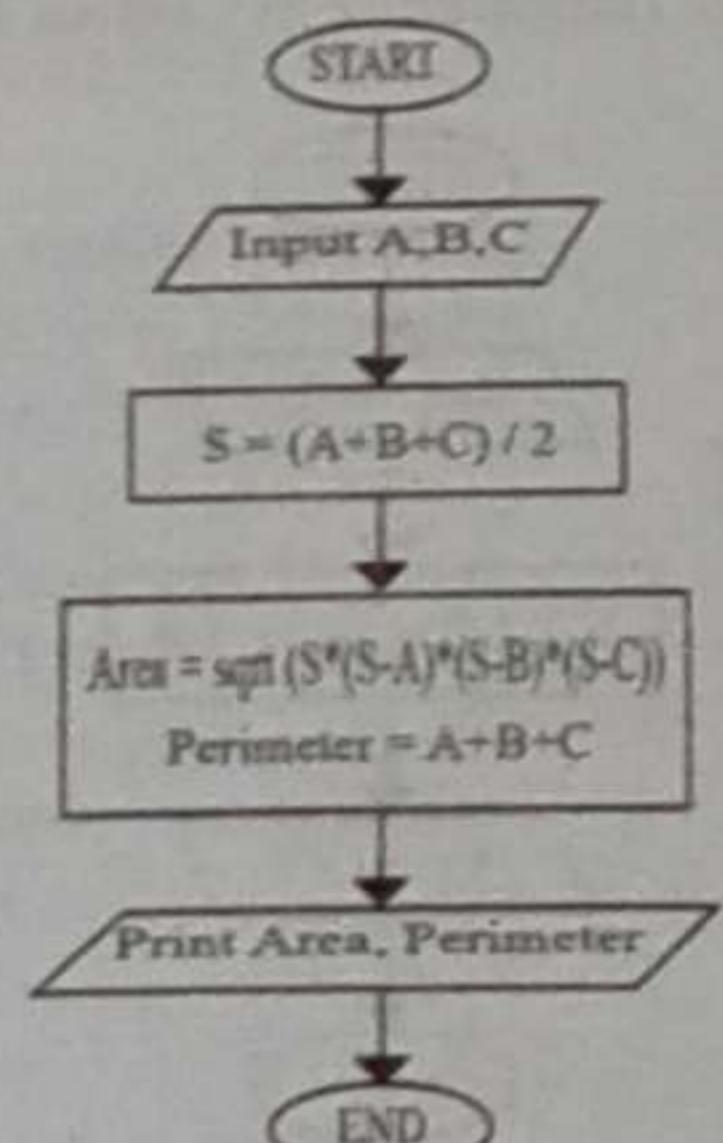
Step-1: Start  
Step-2: Input Radius of Circle say R  
Step-3: Area =  $22.070 * R * R$   
Step-4: PERIMETER =  $2 * 22.070 * R$   
Step-5: Display AREA, PERIMETER  
Step-6: Stop



### 3. Algorithm & Flowchart to find Area & Perimeter of Triangle.

#### Iterative Algorithm:

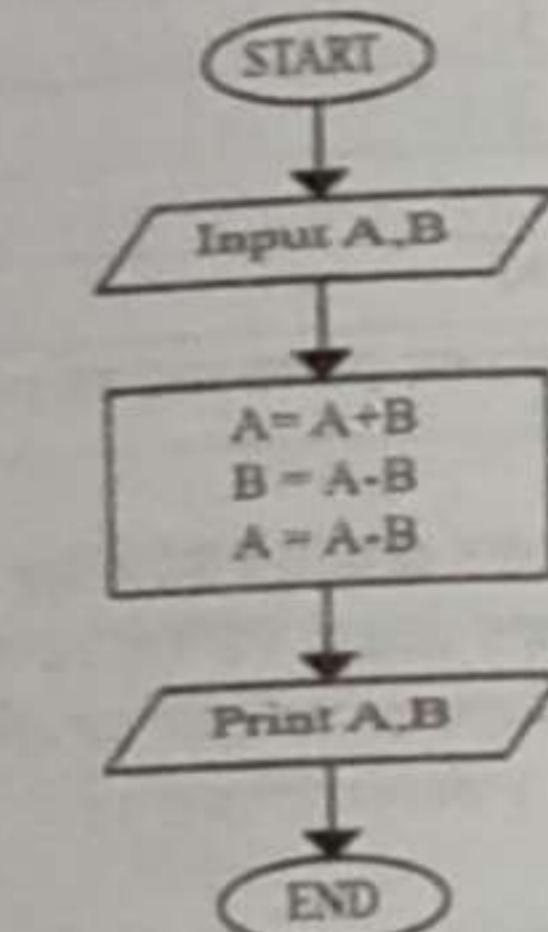
Step-1: Start  
Step-2: Input Sides of Triangle A,B,C  
Step-3:  $S = (A + B + C) / 2$   
Step-4: AREA =  $\sqrt{S * (S-A) * (S-B) * (S-C)}$   
Step-5: PERIMETER =  $A + B + C$   
Step-6: Display AREA, PERIMETER  
Step-7: Stop



### 4. Algorithm & Flowchart to Swap Two Numbers without using temporary variable.

#### Iterative Algorithm:

Step-1: Start  
Step-2: Input Two Numbers Say A, B  
Step-3: Display Before Swap Values A, B  
Step-4:  $A = A + B$   
Step-5:  $B = A - B$   
Step-6:  $A = A - B$   
Step-7: Display After Swap Values A, B  
Step-8: Stop

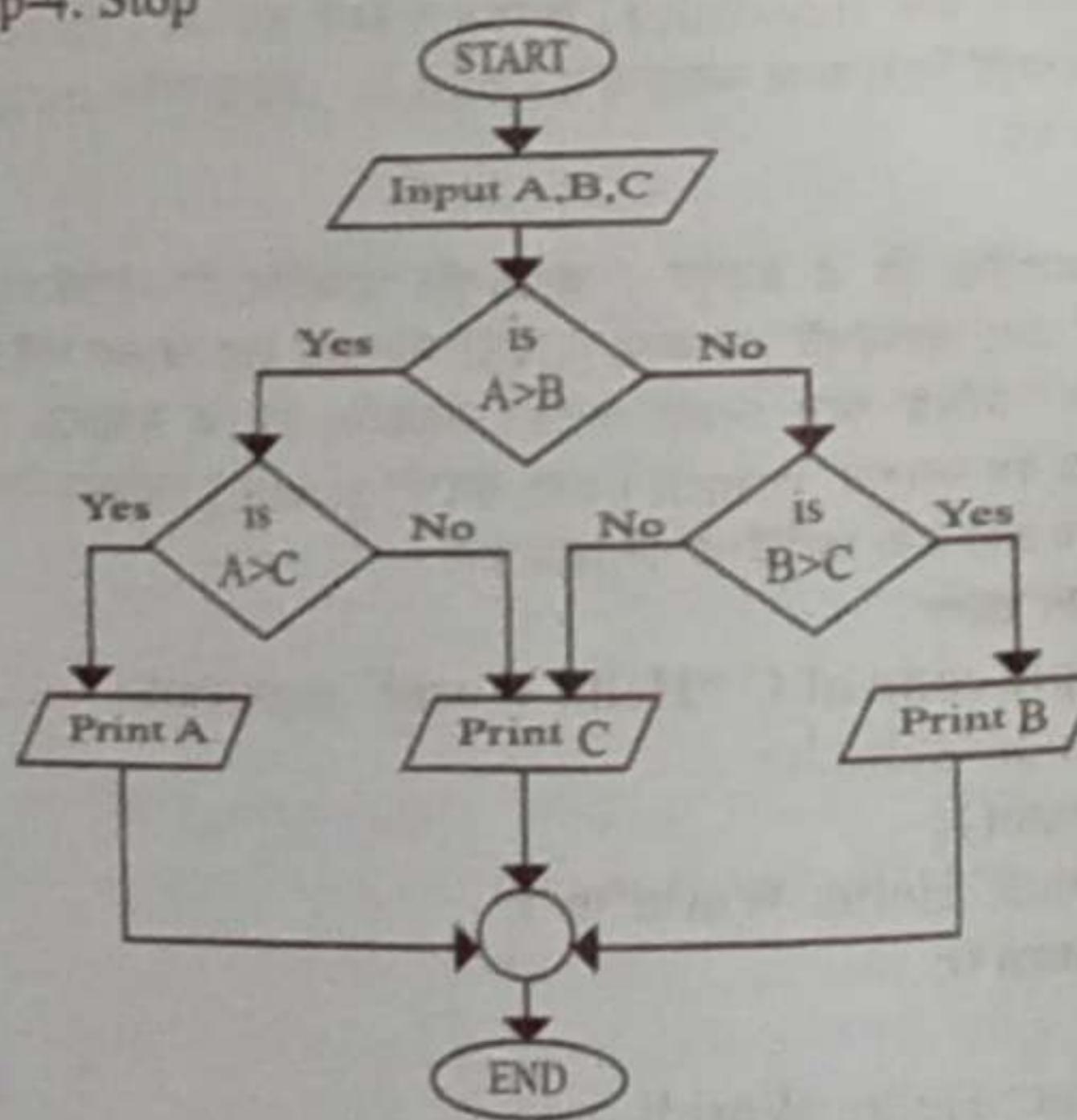


### 5. Algorithm & Flowchart to find the largest of three numbers

#### Iterative Algorithm:

Step-1: Start  
Step-2: INPUT A,B, C  
Step-3: IF A>B AND A>C THEN  
          PRINT A is largest  
ELSE IF B>A AND B>C THEN  
          PRINT B is largest  
ELSE  
          PRINT C is largest  
ENDIF

Step-4: Stop

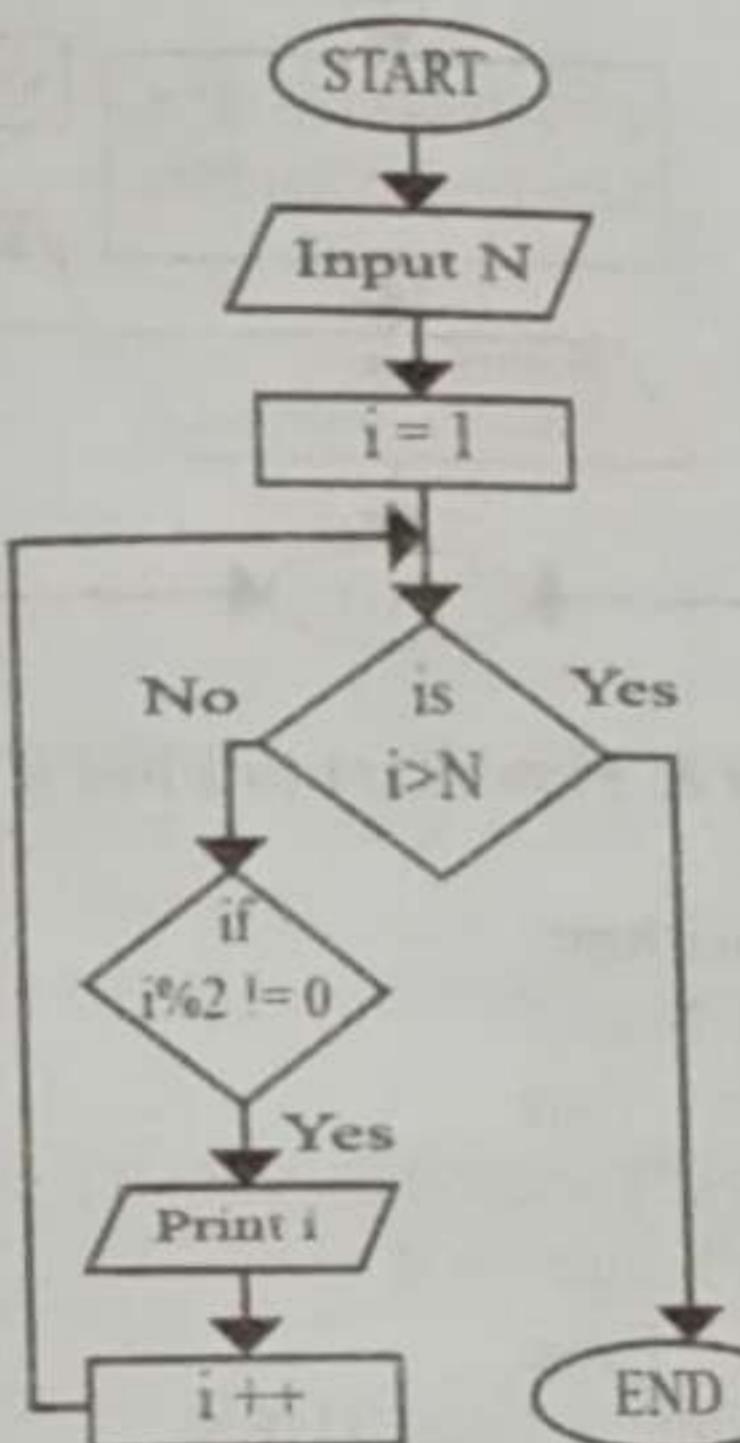


### 6. Algorithm & Flowchart to find Odd numbers between 1 to n where n is a positive Integer

#### Iterative Algorithm:

Step-1: Start  
Step-2: Input Value of N, i=1  
Step-3: While( $i \leq N$ )  
          IF( $i \% 2 \neq 0$ ) THEN  
              PRINT i as Odd number  
               $i++$

Step-4: Stop



### 7. Algorithm & Flowchart to find sum of series $1+2+3+\dots+N$ [NTRCA-2014]

#### Iterative Algorithm:

Step-1: Start  
Step-2: Input Value of N  
Step-3:  $i = 1, SUM=0$   
Step-4: While( $i \leq N$ )  
           $SUM = SUM + i$   
           $i++$

Step-5: Display value of SUM

Step-6: Stop

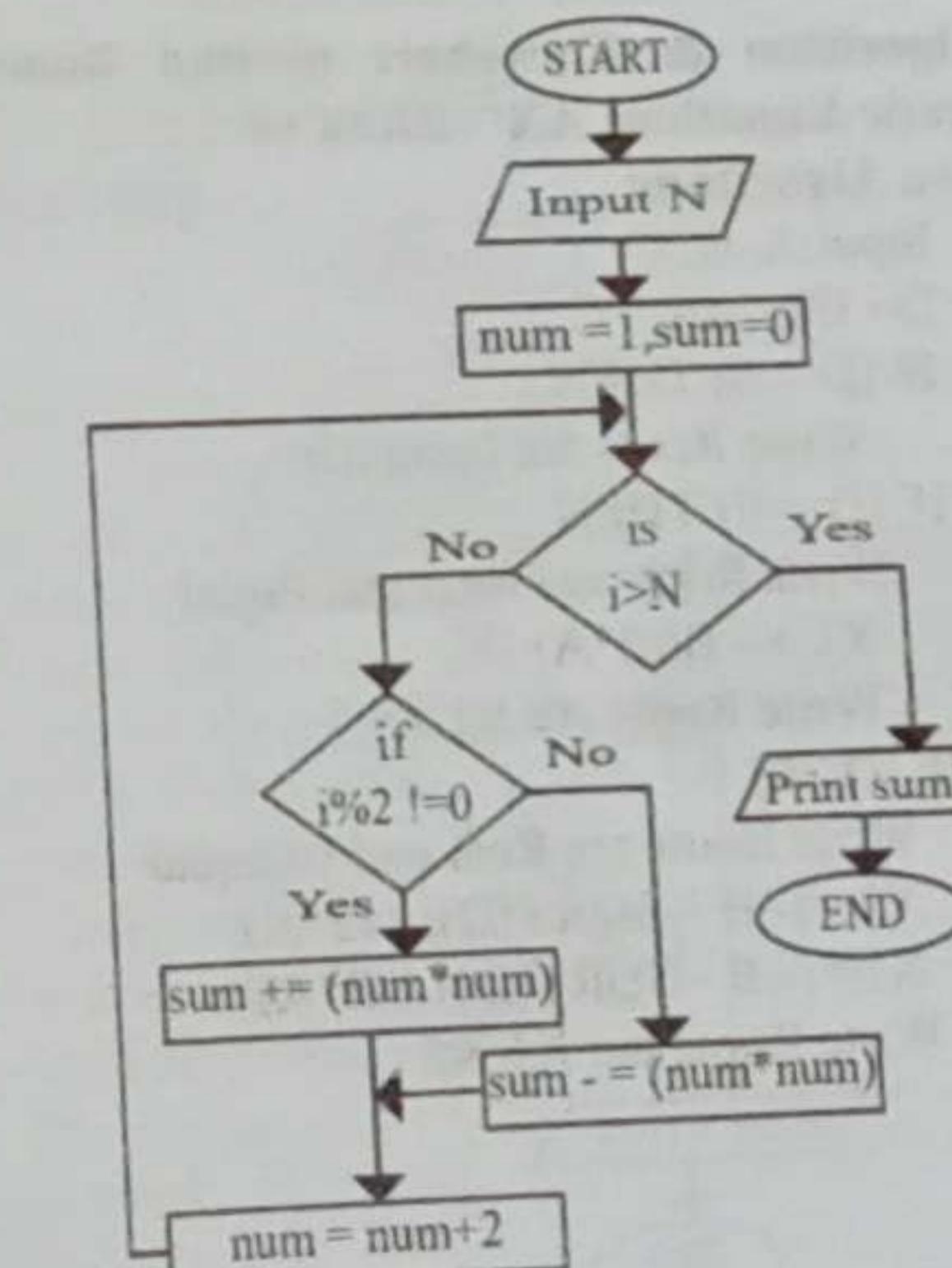
### 8. Algorithm & Flowchart to find sum of series $1^2 - 3^2 + 5^2 - \dots + N^2$

#### Iterative Algorithm:

Step-1: Start  
Step-2: input N  
Step-3: num = 1  
Step-4: Loop:  $i = 1$  to N  
          IF  $i \% 2 \neq 0$  THEN  
              SUM := SUM + num\*num;  
          ELSE  
              SUM := SUM - num\*num;  
              num += 2

Step-5: PRINT SUM

Step-6: Stop



### Structure Programming (C Program)

9. Algorithm & Flowchart to generate first n Fibonacci terms 0,1,1,2,3,5...n (n>2)

Iterative Algorithm: (A = 0, B = 1, C, count=0)

Step-1: Start

Step-2: INPUT n

Step-3: Print A, B

Step-4: WHILE (count<n-1)

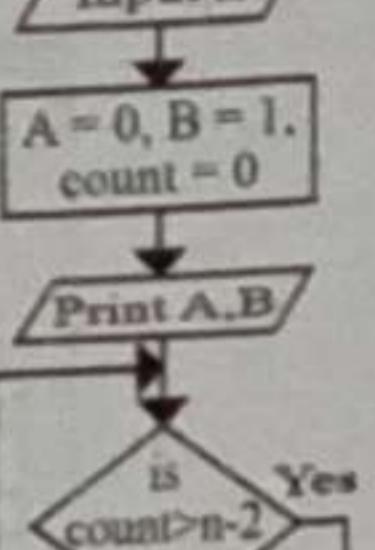
Print C = A+B

A=B

B=C

Count++

Step-5: End



11. Algorithm & Flowchart to Find if a number is prime or not.

Iterative Algorithm:

Step-1: INPUT Num

Step-2: IsPrime = True

Step-3: WHILE(PFactor: 2 to Num-1)  
IF Num % PFactor == 0

THEN IsPrime = False

Step-4: IF (IsPrime = True) THEN  
Display Num is prime

ELSE  
Display Num is not prime

Recursive Algorithm:

Step-1: IsPrime(N,i)

Step-2: Begin

IF i = 1 THEN

RETURN 1;

ELSE

IF N%i = 0 THEN  
RETURN 0

ELSE

Call IsPrime(N, i-1);

ENDIF

ENDIF

Step-3: End

10. Algorithm & Flowchart to find Roots of Quadratic Equations  $AX^2+BX+C=0$

Iterative Algorithm:

Step-1: Input A, B, C

Step-2: D= B<sup>2</sup> - 4A \* C

Step-3: IF (D < 0) THEN

Write Roots are Imaginary

Step-4: IF (D==0) THEN

Write Roots are Real and Equal

$X_1 = -B/(2*A)$

Write Roots are X1,X1

Step-5: IF (D > 0)

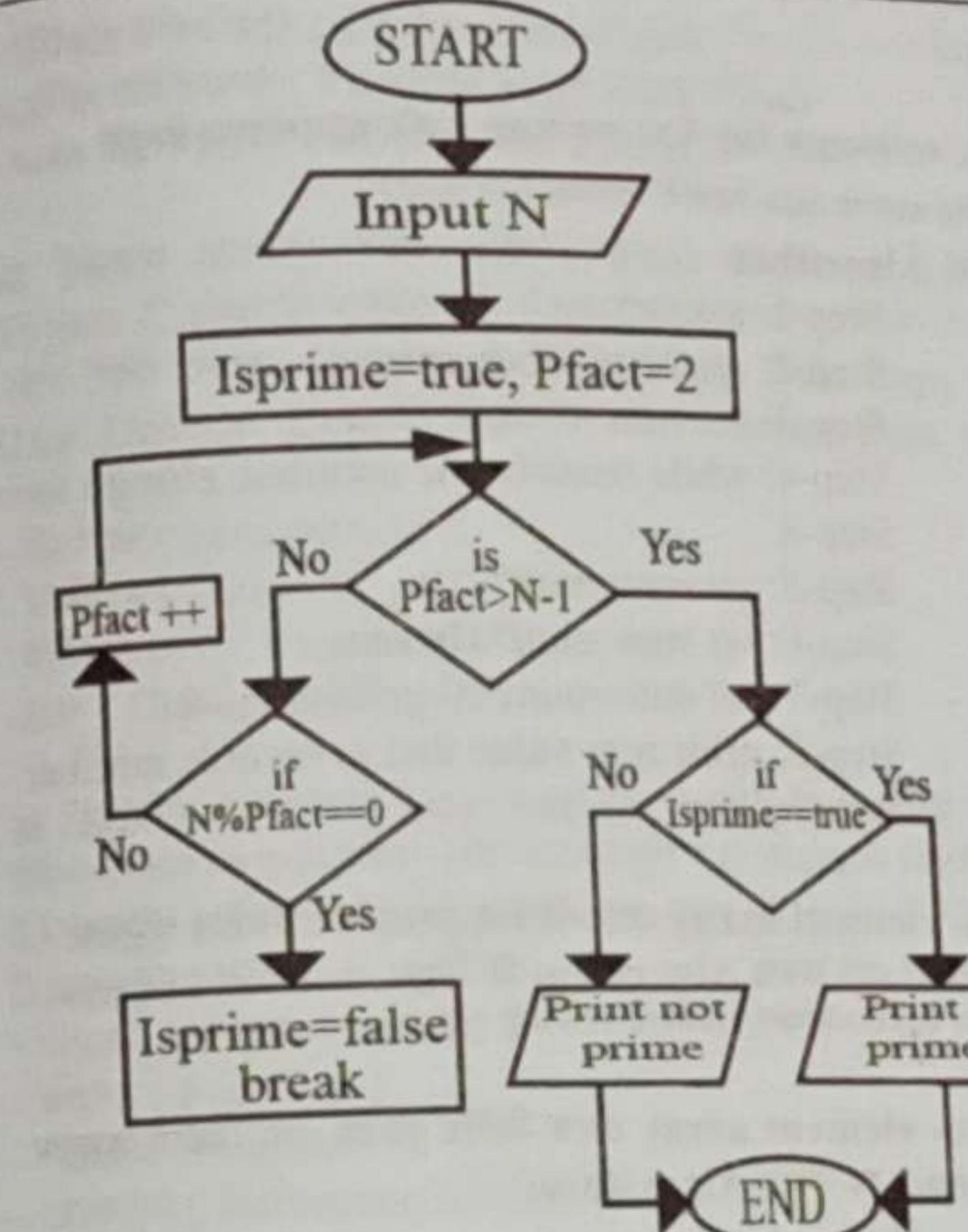
Write Roots are Real and Unequal

$X_1 = (-B + \sqrt{D}) / (2*A)$

$X_2 = (-B - \sqrt{D}) / (2*A)$

Write Roots are X1,X2

### Structure Programming (C Program)



12. Algorithm & Flowchart to Find GCD and LCM of two numbers.

Iterative Algorithm:

Step-1: Start

Step-2: Read two number A, B

Step-3: WHILE (count =1 to A and B)  
IF (A% count and B% count == 0)

THEN gcd = count;

count++;

Step-7 lcm = (a\*b)/gcd

Step-8 Display gcd, lcm

Step-9 Stop

Recursive Algorithm:

Step-1: GCD(A,B)

Step-2: Begin

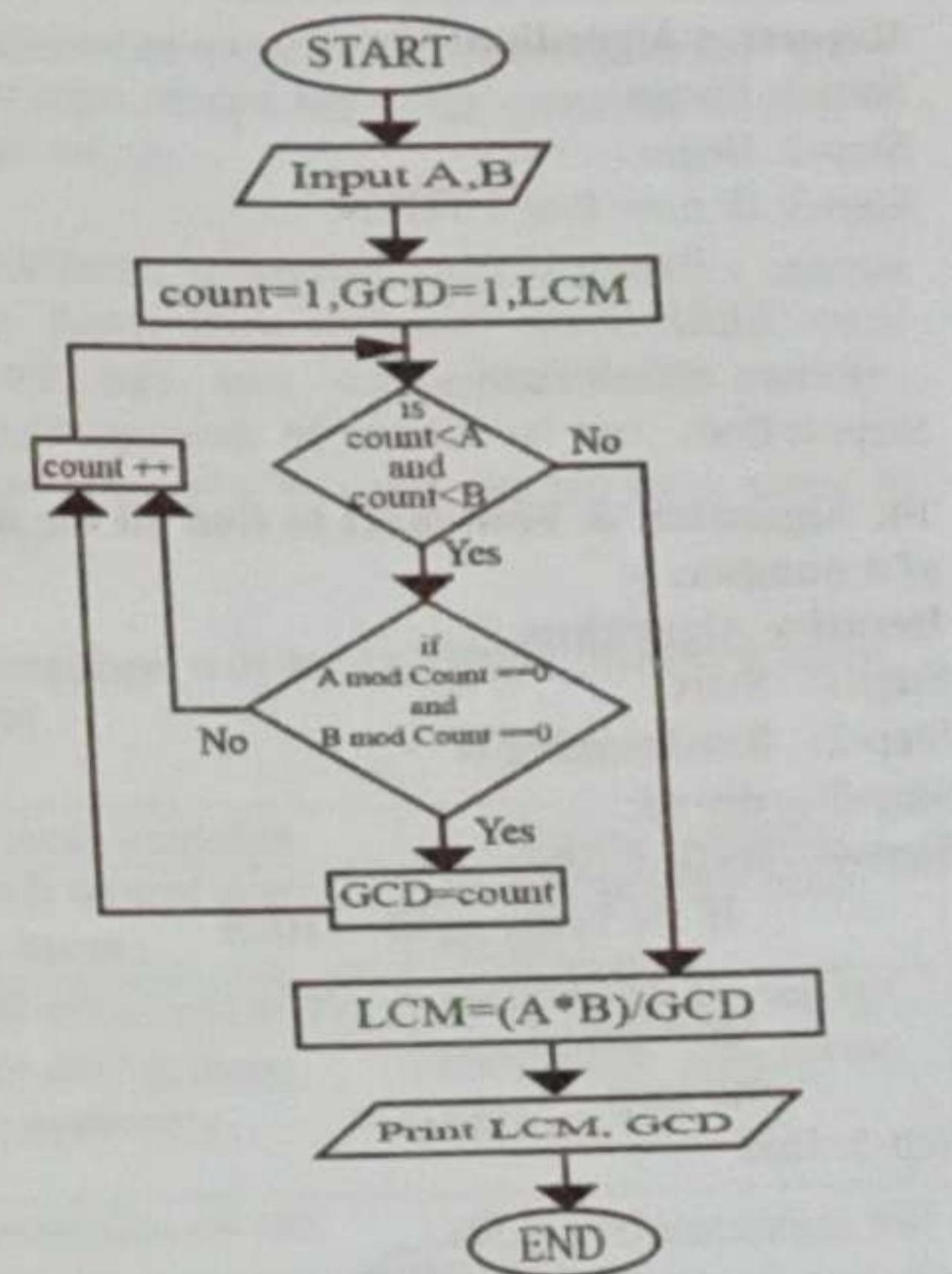
IF B = 0 THEN

RETURN A;

ELSE

Call GCD(B, A%B);

Step-3: End



13. Algorithm & Flowchart to find Factorial of number N ( $n! = 1*2*3*...*n$ ).

Iterative Algorithm:

Step-1: Start

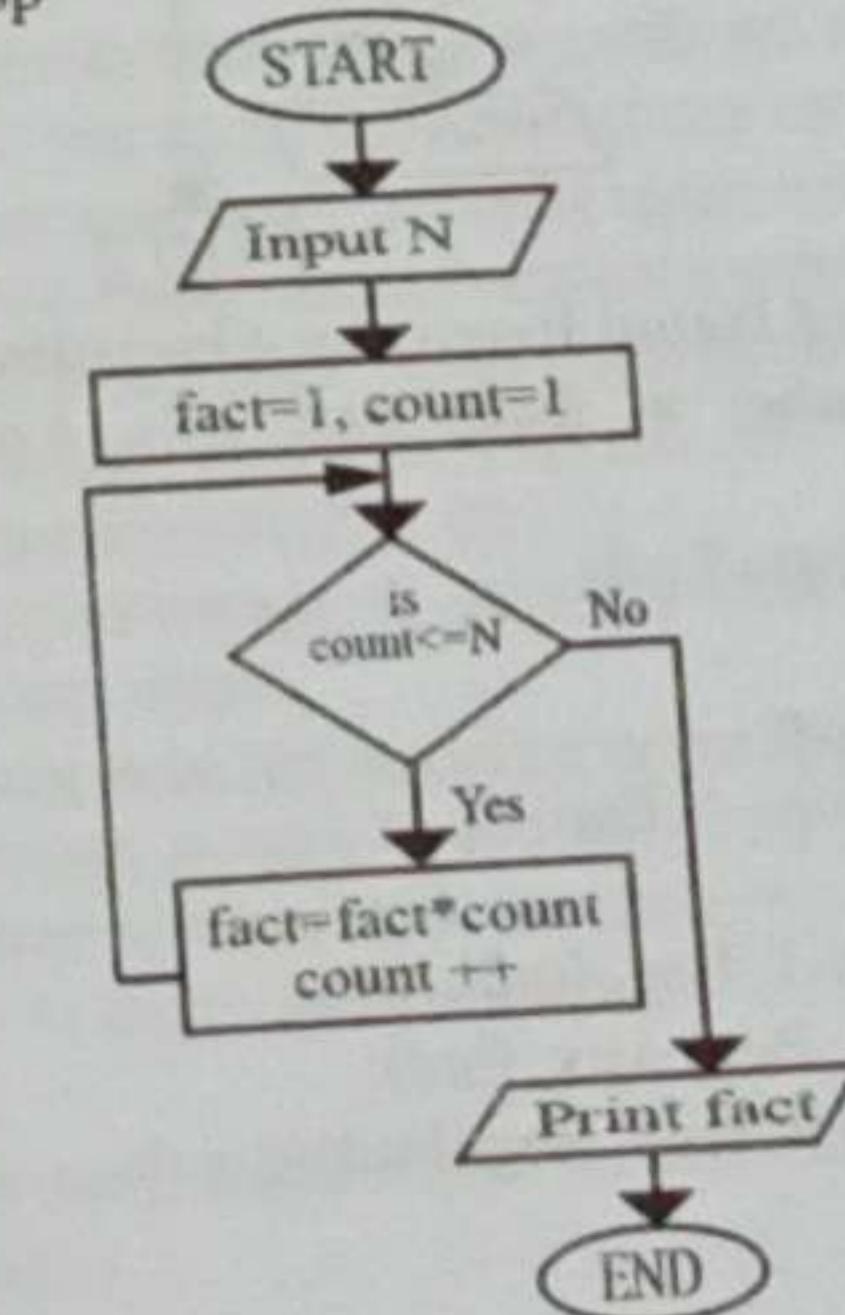
Step-2: Read number N

Step-3: FACT=1, count=1

Step-4: WHILE (count <= N)  
FACT=FACT\* count  
count = count + 1

Step-5: Display FACT

Step-6: Stop



**Recursive Algorithm:**

```

Step-1: Fact(n)
Step-2: Begin
Step-3: IF n == 0 or 1 THEN
    Return 1;
ELSE
    Return n*Call Fact(n-1);
Step-4: End

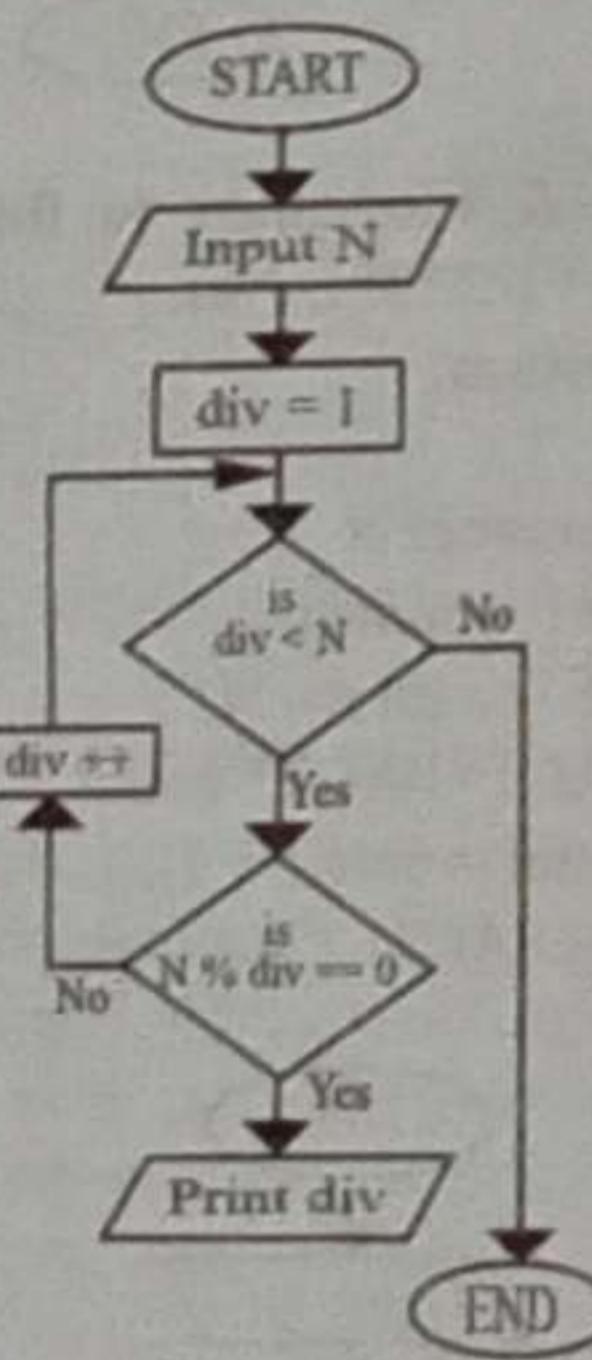
```

**14. Algorithm & Flowchart to find all the divisor of a number.****Iterative Algorithm:**

```

Step-1: Start
Step-2: Read number N
Step-3: div = 1
Step-4: WHILE (div < N)
        IF (N % div == 0) THEN
            PRINT div
            div = div + 1
        End loop
Step-5: End

```

**15. Tower of Hanoi Recursive Algorithm**

```

/*N = Number of disks Beg, Aux, End are the pegs*/
T(N, Beg, Aux, End)
Begin
    if N = 1 then
        Print: Beg --> End;
    else
        Call T(N-1, Beg, End, Aux);
        Call T(1, Beg, Aux, End);
        Call T(N-1, Aux, Beg, End);
    endif

```

End

16. প্যালিনড্রোম নম্বর খুঁজে বের করার একটি অ্যালগরিদম লিখুন।  
[বিভিন্ন মহানাম-২০১৯-সহকারী প্রয়োগার (নন-ক্যাটার)]

**উত্তর: Algorithm:**

```

Step-1: start
Step-2: read a number, num
Step-3: set sum=0, dup=num
Step-4: while num>0 true continue else go to Step-8
Step-5: set rem=num%10
Step-6: set sum=sum*10+rem
Step-7: set num=num/10 go to Step-4
Step-8: print sum value that is reverse number
Step-9: Stop

```

17.n- element array থেকে বিটীয় সূচনাম এবং বিটীয় সূচনাম element বের করার Algorithm টি লিখুন। [তথ্য ও যোগাযোগ প্রক্রিয়ালত ২০১৭-নেটওর্ক ইঞ্জিনিয়ার ও সহকারী প্রয়োগার (নন-ক্যাটার)]

**উত্তর: n- element array থেকে বিটীয় সূচনাম এবং বিটীয় সূচনাম element বের করার Algorithm:**

```

Step-1: Start
Step-2: INPUT ARRAY ELEMENT
Step-3: FOR I TO N
        IF(A[I]<FIRSTMAX &&
        A[I]>SECONDMAX) SECONDMAX=A[I];
        IF(A[I]>FIRSTMIN &&
        A[I]<SECONDMIN) SECONDMIN=A[I];
        PRINT SECONDMAX,
        SECONDMIN
    ENDIF
    ENDIF
Step-4: Stop

```

**Data Type**

■ ডাটা টাইপ: ডাটা টাইপ হল অনেক তলো মানের সমষ্টি, যাহা একটি নির্ধারিত মানের মতো আচরণ করে।

**C Data Types: [NTRCA-2014,2010,14, Multiple Ministry-2017]**

1. Primary Data Types
2. Derived Data Types
3. User Defined Data Type

■ Primary Data Types বা ফার্মেন্টাল ডাটা টাইপ ANSI কৃত্তি নির্ধারিত ডাটা টাইপ সমূহকে বিল্ট ইন ডাটা টাইপ বা আইমির ডাটা টাইপ বা ফার্মেন্টাল ডাটা টাইপ বলে। বিল্ট ইন ডাটা টাইপ ( Built In Data ) সাধারণত চার প্রকার। যথাক্রমে ক্যারেক্টার ডাটা টাইপ (Character Data Type)

**হান্টার টাইপ ডাটা ( Integer Data Type)**

- ফ্লোটিং টাইপ ডাটা ( Floating point Data Type)
- ডাবল টাইপ ডাটা ( Double Data Type)

c. সি ভারিয়েবল এর ব্যবহার ও ভেরিয়েবল ব্যবহার এর প্রধান উদ্দেশ্য হল মেমোরিতে স্টোরকৃত তথ্য যা পরে ব্যবহার করা যায়, এবং যা পরিবর্তন করা যায়।

d. আইডেন্টিফায়ার (Identifier) [NTRCA-2014] : প্রোগ্রামে ব্যবহৃত কোনো নির্দিষ্ট নামকে আইডেন্টিফায়ার বলা হয়। আইডেন্টিফায়ার একটি বর্ণ, একাধিক বর্ণ অথবা একাধিক বর্ণ ও অন্তরের সমন্বয়ে গঠিত হয়। তবে প্রথম ক্যারেক্টার অবশ্যই বর্ণ হতে হবে।

■ C language এ লোকাল এবং গ্লোবাল ভেরিয়েবলের মধ্যকার পার্শ্ব লিখ।

**উত্তর:**

Local Variable	Gobal Variable
এটি একটি ফাংশনের ভিতরে ঘোষণা করা হয়।	এটি ফাংশনের বাইরে ঘোষণা করা হয়।
যদি এটি initialized না করা হয় তবে একটি garbage value সংরক্ষণ করে।	এটি initialized না করা হলে ডিফল্ট হিসাবে zero সংরক্ষণ করে।
এটি execution এর সময় তেরি হয় এবং execution শেষ হয়ে গেলে তা হারিয়ে যায়।	এটি global execution ভুক্ত হওয়ার আগে এবং execution শেষ হয়ে গেলে তা হারিয়ে যায়।
এটি কেবল একটি ফাংশন দ্বারা আয়োজন করা যায় বলে ডেটা ভাগ করা সম্ভব নয়।	একাধিক ফাংশন একই গ্লোবাল ভেরিয়েবল আয়োজন করতে পারে বলে ডেটা ভাগ করা সম্ভব।

প্রশ্ন ২. Variable/ আইডেন্টিফায়ার ঘোষণার নিয়মাবলী লিখ। [NTRCA-2018]

উত্তর: আইডেন্টিফায়ার ঘোষণার নিয়মাবলী সমূহ-

a. আইডেন্টিফায়ারে a হতে z পর্যন্ত হোট হাতের বর্ণ অথবা বড় হাতের বর্ণ, 0 থেকে 9 পর্যন্ত অক্ষ এবং আভার কোর ( \_ ) ব্যবহার করা যাবে।

b. অবশ্যই নামের প্রথম ক্যারেক্টার a হতে z পর্যন্ত হোট হাতের বর্ণ অথবা A হতে Z পর্যন্ত বড় হাতের বর্ণ এর মধ্যে হতে হবে।

c. আইডেন্টিফায়ার শব্দের অর্থগত ক্যারেক্টার সমূহের মাঝে কোন স্পেস বা ফাঁকা রাখা যাবেনা, প্রয়োজনে আভারকোর ব্যবহার করা যাবে।

d. বড় হাতের বর্ণ ও ছোট হাতের বর্ণ ডিম্ব ক্যারেক্টার হিসাবে কাজ করে। সূতরাং Total, total এবং TOTAL শব্দ ডিনটি আলাদা আইডেন্টিফায়ার হিসাবে বিবেচিত হবে।

e. সি ভাষার সংরক্ষিত শব্দসমূহ (Keywords) আইডেন্টিফায়ার হিসাবে ব্যবহার করা যাবেন।

f. আইডেন্টিফায়ার শব্দের মোট ক্যারেক্টার সংখ্যার সীমাবদ্ধতা নাই তবে কম্পাইলার প্রথম ৩১ বর্ণকে বিবেচনা করে।

g. আইডেন্টিফায়ার শব্দে কোন বিশেষ চিহ্ন (+, -, !, @, #, \$, %, ^, &, \*) ইত্যাদি ব্যবহার করা যাবেন।

[ উপরোক্ত নিয়মাবলীয়ার ভেরিয়েবল, ফাংশন, এবং ইত্যাদির নাম ঘোষনা করা যাবে। ]

**■ C তে Variable declare করার Syntax :**

```
Data type variable_name;
or
data type variable_name, variable_name,
variable_name;
```

**■ Variable Definition and Initialization**

Example:

```
int width, height=5;
char letter='A';
float age, area;
double d;
/* actual initialization */
width = 10;
age = 26.5;
```

**Variable Assignment:** একটি variable & value assign  
করার পদ্ধতিকে variable Assignment বলে।

Example:

```
int width = 60;
int age = 31;
```

**■ ডাটা টাইপ ও Variable Declarations in C**

Example:

```
#include <stdio.h>
int main()
{
    int a = 4000; // positive integer data type
    float b = 5.2324; // float data type
    char c = 'Z'; // char data type
    long d = 41657; // long positive integer data type
    long e = -21556; // long -ve integer data type
    int f = -185; // -ve integer data type
    short g = 130; // short +ve integer data type
    short h = -130; // short -ve integer data type
    double i = 4.1234567890; // double float data type
    float j = -3.55; // -ve float data type
}
```

**প্রশ্ন ৩. কোপ ও লাইফ টাইম বলতে কী বুঝায় ?**

উত্তরঃ কোপ কোন ভেরিয়েবলকে প্রয়ামের কোথায় ব্যবহার করা যাবে  
কিংবা কোথায় ব্যবহার করা যাবে না তা নির্ধারণ করে দেয়ার প্রক্রিয়া  
হচ্ছে এই ভেরিয়েবলের কোপ (Scope)।

**■ লাইফ টাইম কোন Variable তার মধ্যে রক্ষিত ভ্যালু (Value)  
কে যত সময় সংরক্ষণ করতে পারে তাই হল এই ভেরিয়েবলের লাইফ  
টাইম।**

**প্রশ্ন 8. Storage class কি ?**

উত্তরঃ একটি স্টোরেজ ক্লাস বলতে বুঝায় একটি সি প্রোগ্রামের মধ্যে  
ভেরিয়েবল অথবা ফাংশন এর নির্দিষ্ট Scope এবং Life time।

**প্রশ্ন ৫. কীওয়ার্ড কি ?**

উত্তরঃ সি প্রোগ্রাম ভাষায় কীওয়ার্ড হচ্ছে সংরক্ষিত কিছু শব্দ মনে রেখে  
একটি বিশেষ অর্থ প্রকাশ করে। সি ৮৯ এ এরকম ৩২ টি কীওয়ার্ড  
আছে।

Auto	break	case	char
Const	while	volatile	continue
Default	do	double	else
Enum	extern	float	for
Goto	if	int	long
Register	return	short	signed
Sizeof	static	struct	switch
Typedef	union	unsigned	void

সি 99 এ এরকম আরো পাঁচটি কীওয়ার্ড যুক্ত হয়।

Bool	Complex	Imaginary	inline	restrict
------	---------	-----------	--------	----------

**প্রশ্ন 11 এ যুক্ত হয় আরো আটটি কীওয়ার্ড।**

Alignas	Alignof	Atomic	Generic
Noreturn	Static	assert	Thread

**প্রশ্ন 6. টোকেন কি ?**

উত্তরঃ সি প্রোগ্রামের সবচেয়ে সূচিতম একটি কে টোকেন বলে।

- 1.Keyword
- 2.Constant
- 3.Identifier
- 4.String
- 5.Operator
- 6.Special Symbol

**প্রশ্ন 7. স্টেটমেন্ট এবং এক্সপ্রেশন কি ?**

উত্তরঃ স্টেটমেন্ট কতগুলো টোকেন থখন একসাথে মিলে প্রয়ামে  
একটি একক কাজ সম্পন্ন করে তখন সেই টোকেনগুলোকে একসাথে  
স্ট্যাটমেন্ট বলে।

এক্সপ্রেশন থখন কোন গানিতিক টার্ম কোন স্টেটমেন্ট এ থাকে তখন  
তাকে স্টেটমেন্ট না বলে এক্সপ্রেশন বলতে হয়।

**প্রশ্ন 8. C তে ব্যবহৃত বিভিন্ন Formate Specifier লেখ।**

Formate Specifier এর নাম	ব্যবহার
%c	তধু মাত্র একটি অক্ষর / character I/O এর জন্য।
%d	প্রকৃত সংখ্যা (integer) ডাটা টাইপের ভ্যালু I/O এর জন্য।
%e	দশমিক যুক্ত (Floating) ডাটা টাইপের সংখ্যা exponential notation I/O এর জন্য।
%E	দশমিক যুক্ত (Floating) ডাটা টাইপের সংখ্যা exponential notation I/O এর জন্য।
%f	দশমিক যুক্ত নাখার (Floating) ডাটা টাইপ ভ্যালু I/O এর জন্য।
%g	দশমিক যুক্ত (Floating) ডাটা টাইপের সংখ্যা কে %f অথবা %g তাবে প্রদর্শনের জন্য।
%G	দশমিক যুক্ত (Floating) ডাটা টাইপের সংখ্যা কে %G তাবে I/O এর জন্য।

%o	অক্টাল ডাটা টাইপের ভ্যালু I/O এর জন্য।
%s	Character String ডাটা I/O এর জন্য।
%u	Unsigned ইন্টেজার I/O এর জন্য।
%x	ছোট অক্ষরের হেক্সাডেসিমেল ভ্যালু I/O এর জন্য।
%X	বড় অক্ষরের হেক্সাডেসিমেল ভ্যালু প্রদর্শনের জন্য।
%b	বাইনারি ডাটার জন্য।

বিন্দুত প্রতিটা স্টেটমেন্টের শেষে একটা সেমিকোলন থাকে। তাই মেধানে  
মেধানে সেমিকোলন দেয়া দরকার সেখানে সেমিকোলন না দিলে  
কম্পাইলার "statement missing" এর দেখায়।

**প্রশ্ন 9. Constants কি?**

উত্তরঃ Constants যাদের মান সব সময়ের জন্য ছিল তাদেরকে  
Constant বলে। যেমনং ১ এর মান সবসময় ১ ই থাকবে তাহলে  
Constants কে literals ও বলা হয়। Constants এ কোন  
data type হতে পারে।

**Constant Syntax:**

```
const data_type constant_name;
```

**প্রশ্ন 10. বিভিন্ন ধরনের কনস্ট্যান্ট in C Language.**

উত্তরঃ

- a. Floating point Constant
- b. String constant
- c. Integer Constant etc.

**■ Backslash character constant:** C কিছু special  
backslash '\' character constants সমর্থন করে যা আউটপুট  
ফাংশন ব্যবহার করা হয়।

Constants	Meaning
\f	form feed
\n	new line
\r	carriage return
\t	horizontal tab
\v	vertical tab
\'	single quote
\\"	double quote
\।	backslash
\0	null
\b	back space

**প্রশ্ন 11. Preprocessors কি ?**

উত্তরঃ Preprocessor একটি প্রোগ্রাম যা কম্পাইলার দ্বারা call করা  
হয়। '#' symbol দিয়ে pre-processor directives লিখা হয়।

**প্রশ্ন 12. সি প্রোগ্রাম এ ব্যবহৃত বিভিন্ন Header file ও তার বিবরণ  
দাও।**

উত্তরঃ

Header file	Description
stdio.h	Standard input/output header file. ইহাতে Input/Output functions সমূহ যোগ্য করা থাকে।

conio.h	Console input/output header file
string.h	ইহাতে String related function সমূহ মোহৃণ করা থাকে।
stdlib.h	ইহাতে general standard library function সমূহ মোহৃণ করা থাকে।
math.h	ইহাতে Math related function সমূহ মোহৃণ করা থাকে।
time.h	ইহাতে Time and clock related function সমূহ মোহৃণ করা থাকে।
ctype.h	ইহাতে character handling function সমূহ মোহৃণ করা থাকে।
locale.h	জার্মানিকরণ ফাংশন।
signal.h	সিগনাল হ্যান্ডলিং ফাংশন।
error.h	Error handling functions

**অ সি প্রোগ্রাম অপারেটর কী ?**

উত্তরঃ অপারেটর হল একটি প্রতীক যা এক বা একাধিক ভ্যালুর বা  
ভ্যারিয়েবল এর উপর কাজ করে। উদাহরণস্বরূপ: + একটি অপারেটর যা  
যোগ এর কাজ করে।

**প্রশ্ন 2. ইউনারি অপারেটর কী ?**

উত্তরঃ যে সব অপারেটর একটি মাত্র অপারেট নিয়ে কাজ করে তাদের  
ইউনারি অপারেটর বলে।

**প্রশ্ন 3. বাইনারি অপারেটর কী ?**

উত্তরঃ যে সব অপারেটর দুইটি অপারেট নিয়ে কাজ করে তাদেরকে  
বাইনারি অপারেটর বলে।

**■ Operators In C:**

- a. Arithmetic Operators
- b. Relational Operators
- c. Logical Operators
- d. Increment/Decrement
- e. Assignment Operators
- f. Conditional Operators
- g. Bitwise Operators
- h. Special Operator</li

## Structure Programming (C Program)

### b. Relational Operators :

রিলেশনাল অপারেটর দুইটি অপারেটরের মধ্যের সম্পর্ক চেক করে। যদি সম্পর্ক টি সত্ত্ব হয় তাহলে এটা 1 value রিটার্ন করে এবং যদি মিথ্যা হয় তাহলে 0 value রিটার্ন করে। যেমন:  $a > b$  এখানে > হল একটি রিলেশন অপারেটর। যদি  $a, b$  এর খেকে বড় হয় তাহলে  $a > b$ । রিটার্ন করবে আর যদি তা না হয় তাহলে 0 রিটার্ন করবে। প্রোগ্রামিং সি তে ডিসিশন মেকিং এবং মূল এ রিলেশনাল অপারেটরের ব্যবহার করা হয়।

অপারেটর	উদাহরণ	অর্থ ক্রপ
$=$	Equal to	$5=3$ returns false (0)
$>$	Greater than	$5>3$ returns true (1)
$<$	Less than	$5<3$ returns false (0)
$!=$	Not equal to	$5!=3$ returns true(1)
$>=$	Greater than or equal to	$5>=3$ returns true (1)
$<=$	Less than or equal to	$5<=3$ return false (0)

### c. Logical Operators :

রিলেশন অপারেটরস জোর এক্সপ্রেশন এক্ষত্রিত করার জন্য লজিকাল অপারেটর ব্যবহার করা হয়। প্রোগ্রামিং সি তে 3 টি লজিকাল অপারেটর রয়েছে।

অপারেটর	উদাহরণ	অর্থ ক্রপ
$&&$	Logical AND	If $c=5$ and $d=5$ then, $((c==5) && (d>5))$ returns false.
$  $	Logical OR	If $c=5$ and $d=5$ then, $((c==5)    (d>5))$ returns true.
!	Logical NOT	If $c=5$ then, $!(c==5)$ returns false.

ব্যাখ্যা:  $((c==5) && (d>5))$  এক্সপ্রেশনটি সত্ত্ব হওয়ার জন্য  $c==5$  এবং  $d>5$  এই দুইটা কভিশনই সত্ত্ব হতে হবে, কিন্তু উদাহরণ এ দেওয়া  $d>5$  কভিশনটি মিথ্যা তাই এই এক্সপ্রেশনটি মিথ্যা।  $!(c==5) || (d>5))$  এক্সপ্রেশনটি সত্ত্ব কারণ এই এক্সপ্রেশনটির 2 টা কভিশন সত্ত্ব।  $!(c==5)$  এক্সপ্রেশনটি মিথ্যা কারণ কভিশনে  $c=5$  দেওয়া আছে।

### d. Increment এবং Decrement অপারেটর : [NTRCA-2011]

প্রোগ্রামিং সি তে “ $++$ ” এবং “ $--$ ” কে যথাক্রমে ইনক্রিমেন্ট (Increment) এবং ডিক্রিমেন্ট (Decrement) অপারেটরের হিসেবে ব্যবহার করা হয়। যেমন, টেস্ট কভিশন অনুসারে বিভিন্ন স্টেটমেন্ট এক্সিকিউট হওয়া, এটা সত্ত্ব নাকি মিথ্যা তার উপর নির্ভর করে।

#### কভিশনাল অপারেটরের Syntax :

conditional \_ expression?expression1 : expression2  
যদি টেস্ট কভিশন সত্ত্ব হয় তাহলে এক্সপ্রেশন 1 রিটার্ন করবে আর যদি মিথ্যা হয় তাহলে এক্সপ্রেশন 2 রিটার্ন করবে। উদাহরণ ক্রমপঞ্চাংশ  
`#include <stdio.h>`  
`int main()`

ধরি,  $a=5$  এবং  $b=10$

$a++ ; //a = 6$  হবে

$a-- ; //a = 5$  হবে

```
++a; //a = 6 হবে
--a; //a = 5 হবে
```

■ Post-fix এবং Pre-fix হিসেবে “ $++$ ” এবং “ $--$ ” অপারেটরের মধ্যে পার্থক্য [NTRCA-2014]  
যখন “ $++$ ” প্রি-ফিল্ড হিসেবে ব্যবহার করা হবে, যেমন  $++a$ , তখন আগে  $a$  এর মান 1 বৃক্ষি করবে তারপর  $a$  এর মান দেখাবে। একইভাবে “ $--$ ” যখন পোস্ট-ফিল্ড হিসেবে ব্যবহার করা হবে, যেমন  $a--$ , তখন অপারেটর আগে অপারেট এর মান রিটার্ন করবে এবং তারপর সেটার মান 1 বৃক্ষি করবে। “ $--$ ” অপারেটর ও পোস্ট-ফিল্ড এবং প্রি-ফিল্ড হিসেবে একইভাবে কাজ করে। নিচে একটি উদাহরণ এর সাথেয়ে বিবরণ দুর্বাল।

```
#include <stdio.h>
int main()
{
    int c=2;
    printf("%d\n",c++); //এই স্টেটমেন্টটি প্রথমে c এর
    //মান 2 display করবে তারপর c এর মান 1 বেড়ে 3 হবে।
    printf("%d",++c); //এই স্টেটমেন্টটি প্রথমে c এর মান 1
    //বাড়াবে এবং তারপর সেটা c এর মান display করবে।
    return 0;
}
```

### e. Assignment Operator :

ব্যক্ত ব্যবহৃত এসাইনমেন্ট অপারেটর হল “ $=$ ” অপারেটর। এই অপারেটরটি অপারেটর এর ডান পাশের value এর বাম পাশের variable এ এসাইন করে। উদাহরণস্বরূপ :

```
var=5; //5 is assigned to var
a=c; //value of c is assigned to a
5=c; // Error! 5 is a constant
```

অপারেটর	Short Hand	Long Hand
=	$a=b$	$a=b$
+=	$a+=b$	$a=a+b$
-=	$a-=b$	$a=a-b$
*=	$a*=b$	$a=a*b$
/=	$a/=b$	$a=a/b$
%=	$a\%b$	$a=a\%b$

### f. Conditional Operators :

প্রোগ্রামিং সি তে ডিসিশন মেকিং এর জন্য কভিশনাল অপারেটরস ব্যবহার করা হয়। যেমন, টেস্ট কভিশন অনুসারে বিভিন্ন স্টেটমেন্ট এক্সিকিউট হওয়া, এটা সত্ত্ব নাকি মিথ্যা তার উপর নির্ভর করে।

#### কভিশনাল অপারেটরের Syntax :

conditional \_ expression?expression1 : expression2  
যদি টেস্ট কভিশন সত্ত্ব হয় তাহলে এক্সপ্রেশন 1 রিটার্ন করবে আর যদি মিথ্যা হয় তাহলে এক্সপ্রেশন 2 রিটার্ন করবে। উদাহরণ ক্রমপঞ্চাংশ  
`#include <stdio.h>`  
`int main()`

## Structure Programming (C Program)

### ■ OR অপারেটর:

Logical OR এর মতো কাজ করে। এখন যদি আমরা আবারও আগের দুটি নামার বিবেচনা করি, 78 এবং 45 এদের মধ্যে OR অপারেশন করা হলে X এর মান কি হবে।

```
x = 78 | 45;
78 = 0100 1110
45 = 0010 1101
```

$x = 0110 1111 = 111$

[২ টি binary number এর same হানীয় অংকের সাথে OR operation হবে]

### ■ NOT অপারেটর:

NOT অপারেটরের কাজ হল এটি যে নামারের উপর অপারেশন চালায় সেই নামারের প্রতিটি বিটকে ট্রিপল (complement) করে দেয়। অর্থাৎ যে বিটে 0 আছে তাকে 1 বানায়, আর যে বিটে 1 আছে তাকে 0 বানিয়ে দেয়। আবারও ধরা যাক দুটি নামার 78 এবং 45। এদেরকে বিট-ওয়াইজ অপারেটরের ব্যবহারের মাধ্যমে হিসাব সম্পন্ন করলে অনেক সহজ বৰ্তানো যায়, তথা প্রোগ্রামের efficiency অনেকাংশে বৃক্ষি করা যাব। সি ল্যাঙ্গুজেজে 6 টি বিট-ওয়াইজ অপারেটর রয়েছে, যথে :

অপারেটর	অপারেটরের অর্থ
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Shift left
>>	Shift right

### ■ AND অপারেটর:

বিট-ওয়াইজ AND অপারেটরটি লজিকাল AND ( $&&$ ) এর মতই কাজ করে। পার্থক্য হল, এটি প্রতিটি বিটের উপর আলাদা আলাদাভাবে কাজ করে। ধরা যাক আমাদের কাছে দুটি unsigned int আছে। হিসাবের সুবিধার জন্য আমরা unsigned int এর সাইজ 32 বিটের জয়বায়াম ধরে নিছি 8 বিট। এখন ধরা যাক একটি নামার 78 এবং আরেকটি নামার 45। আমরা যদি নামার দুটিকে বাইনারিতে জপান্ত করি তাহলে তাদের বিট প্যাটার্ন পেয়ে যাব।

$78 = 01001110$

$45 = 00101101$

$x = 0110 0011 = 99$

[২ টি binary number এর same হানীয় অংকের সাথে XOR operation হবে]

### ■ LEFT SHIFT এবং RIGHT SHIFT অপারেটর:

SHIFT অপারেটরগুলো কোন নামারের সময় বিট প্যাটার্নকে ডানে বা বামে শিফট করে দেয়। অর্থাৎ যতক্ষণ বলা হবে এরা নামারের বিটগুলোর প্রতিটিকে বা অবস্থান থেকে ততক্ষণ ডানে বা বামে সরিয়ে দিবে। LEFT SHIFT বামে এবং RIGHT SHIFT অপারেটর ডানে সরাবে। SHIFT অপারেটরের সিট্যাকে অপারেটরের বামে থাকে যে নামারটিকে শিফট করা হবে সেটি এবং ডানে থাকে যতক্ষণ শিফট করা হবে সেই নামার। ধরা যাক আমরা 78 কে 1 ঘর বামে শিফট করতে চাই। তাহলে ব্যাপারটি কিরকম হবে দেখা যাব।

$78 = 0000 0000 0100 1110$

```
0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0
x = 78 << 1;
```

```
0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0
x = 0000 0000 1001 1100 = 156 // একবার বামে শিফট  
করার পর একইভাবে RIGHT SHIFT অপারেশন করা হলে যত্থব  
কলা থাকে নাখারটির বিটগুলো তত ঘর তানে সবে যায়। LEFT  
SHIFT এ যেহেন বামের বিটগুলো হালিয়ে যায়, RIGHT SHIFT এ  
ভাবের বিটগুলো হালিয়ে যায় এবং বামে 0 যোগ হতে থাকে।
```

```
45=0000 0000 0010 1101
```

```
0 0 0 0 0 0 0 0 0 1 0 1 1 0 1
x = 45 >> 2
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1
x = 0000 1011 => 11 // দুইবার তানে শিফট করার পর
```

#### h. Special Operator

- a. Address Operator ("&")
- b. Pointer Operator ("\*")
- c. Comma Operator
- d. size of operator etc.

**Comma Operator :** পরস্পর সম্পর্কযুক্ত একাধিক একাধিক প্রোগ্রাম লিখতে হবে যখন আমরা একসাথে করার জন্য কমা অপারেটর ব্যবহার করা হয়। যেমন, int a,c=5,d;

**Size of Operator :** এটা একটি ইউনারি অপারেটর যা ব্যবহার করে ভেটাটাইপ, কলস্ট্যাক্ট অ্যারে, স্ট্রাকচার সহ ইত্যাদির সাইজ জানার জন্য। উদাহরণস্বরূপ নিচের প্রোগ্রামটি দেওয়া হল।

```
#include <stdio.h>
int main(){
    int a;
    float b;
    double c;
    char d;
    printf("Size of int=%d bytes\n",sizeof(a));
    printf("Size of float=%d bytes\n",sizeof(b));
    printf("Size of double=%d bytes\n",sizeof(c));
    printf("Size of char=%d byte\n",sizeof(d));
    return 0;
}
```

#### আউটপুট :

```
Size of int=4 bytes
Size of float=4 bytes
Size of double=8 bytes
Size of char=1 byte
```

#### Control Flow Statement

মে স্টেটমেন্ট Condition এর উপর ভিত্তি করে decision নিতে  
সাহায্য করে সেই statements কে বলা হয় Decision Making  
Statements or Conditional Statements.

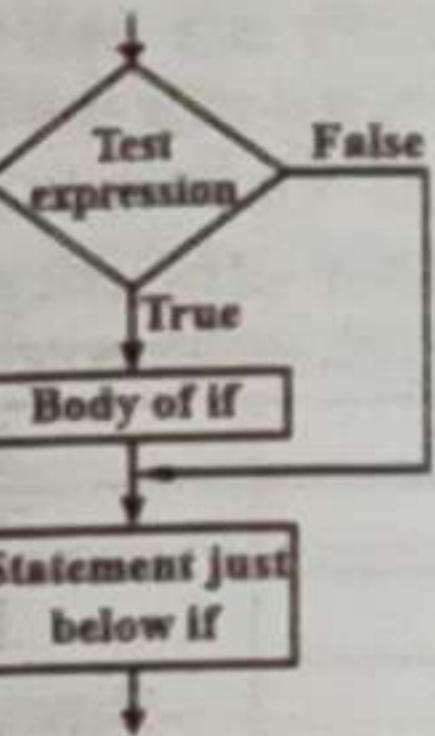
- ভিত্তি conditional statements :
- a. if
- b. if else
- c. else if
- d. Nested if
- e. Switch
- f. break
- g. continue
- h. goto

**a. if Statement:** if স্টেটমেন্ট প্রথমে একাধিক চেক করবে। যদি  
এটি সত্য হয় তাহলে এটি এক্সিকিউট করবে, যিন্তা হলে if স্টেটমেন্ট  
এর মধ্যে কি আছে তা প্রোগ্রাম কর্তৃপক্ষ চেক করবে না।

#### Syntax :

```
if (test expression) {
    // স্টেটমেন্টটি কাজ করবে যখন টেস্ট একাধিক সত্য হবে
}
//
```

#### Flow chart:



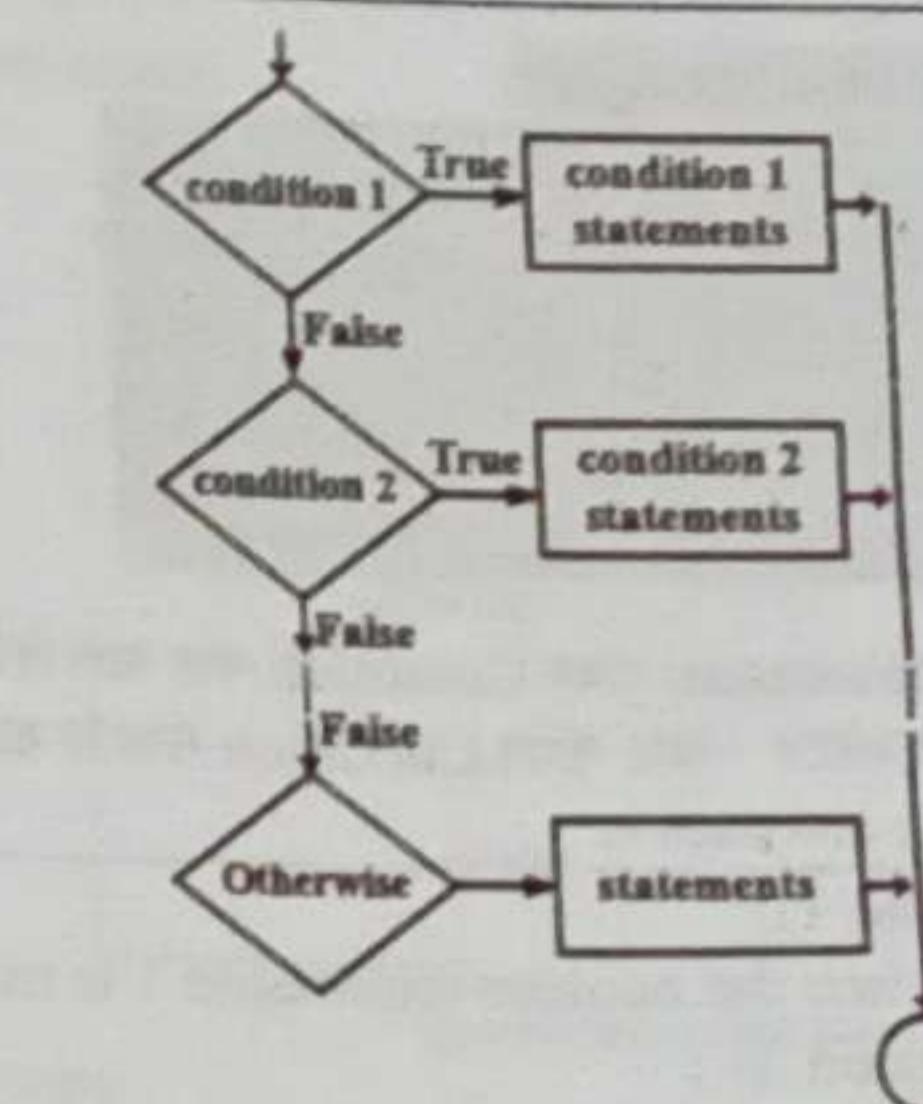
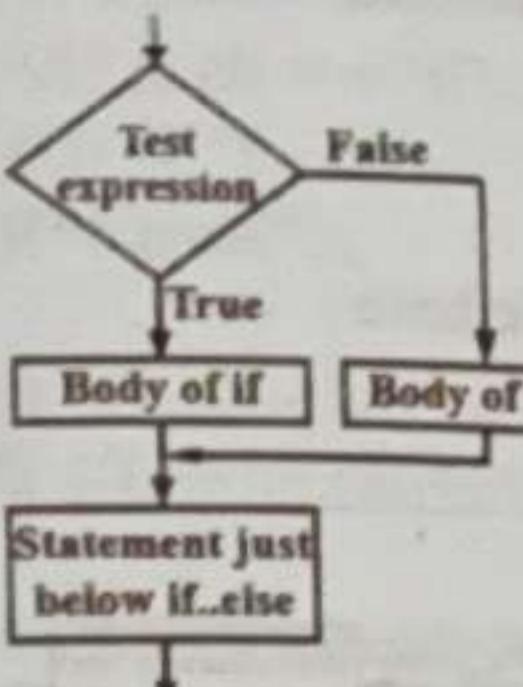
এখন আমরা একটি প্রোগ্রাম এর মাধ্যমে If স্টেটমেন্ট এর উদাহরণ  
দেখতে পারি। এমন একটি প্রোগ্রাম লিখতে হবে যে কোন নামের ইনপুট  
করলে সেটা ইন্ডেন না অত (জোড় না বিজোড়) সেটা Output এ  
দেখাবে।

```
#include<stdio.h>
int main(){
    int num;
    scanf("%d",&num);
    if((num%2)==0) // চেক করছে তাগশেষ 0 কিনা//
        printf("%d is even.",num);
    else
        printf("%d is odd.",num);
    return 0;
}
```

**c. else if Statement:** যখন একটি প্রোগ্রাম একের অধিক টেস্ট  
একাধিক ব্যবহার করা হয় তখন else if স্টেটমেন্ট ব্যবহার করা হয়।  
else if স্টেটমেন্ট এর সিনট্যাক্সিঃ

```
if (test expression 1) {
    // স্টেটমেন্টটি কাজ করবে যখন টেস্ট একাধিক সত্য হবে //
}
else if(test expression 2) {
    // স্টেটমেন্টটি কাজ করবে যখন টেস্ট একাধিক সত্য হবে //
}
else if (test expression 3){
    // স্টেটমেন্টটি কাজ করবে যখন টেস্ট একাধিক সত্য হবে //
}
else {
    // স্টেটমেন্টটি কাজ করবে যখন টেস্ট একাধিক সত্য হবে //
}
```

#### Flow chart:



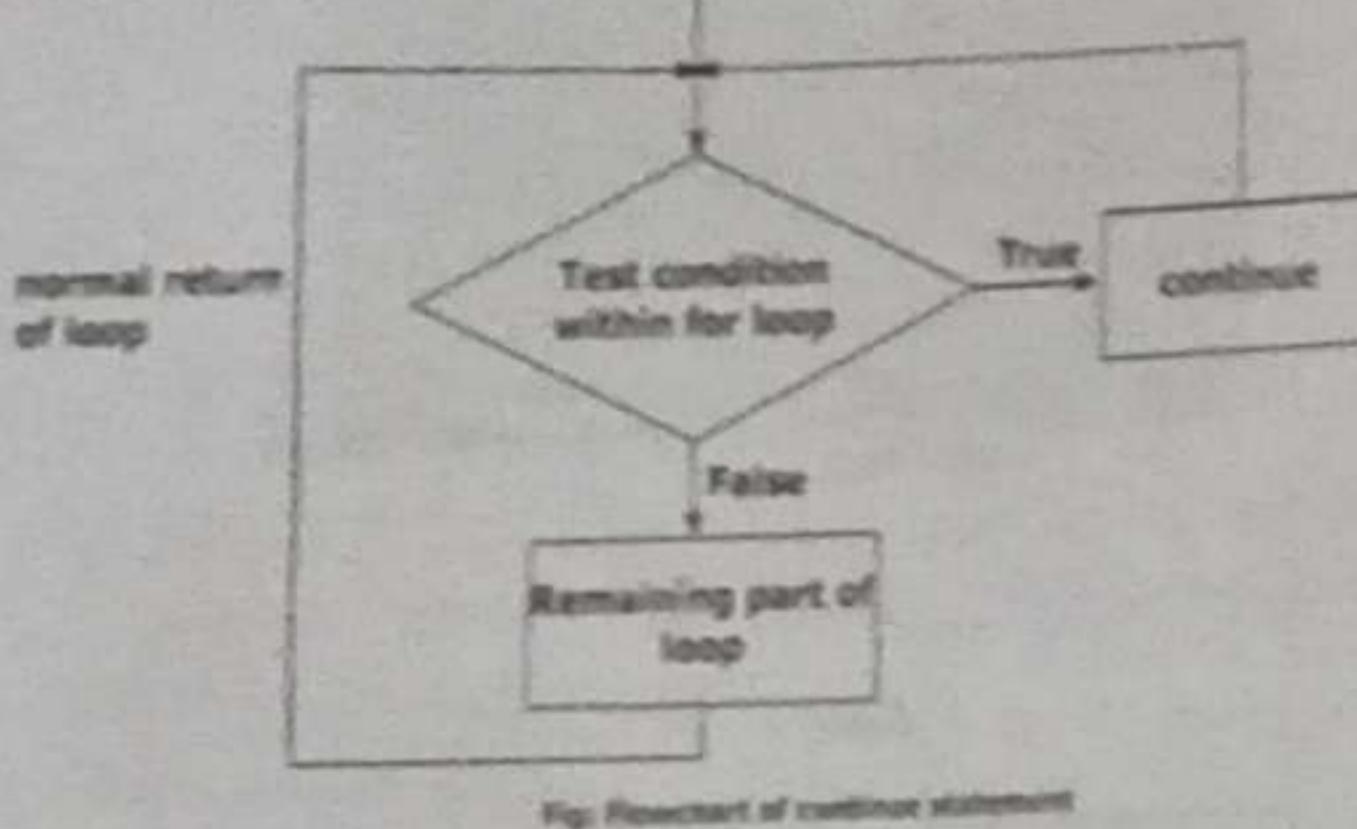
উদাহরণঃ Write down a program in C language that  
will find the maximum of four integers given as  
inputs. [Ministry of Agriculture-SAE-2021]

#### উৎস: Sample Code:

```
#include<stdio.h>
int main() {
    int a,b,c,d;
    printf("Enter the Four Numbers:\n");
    scanf("%d %d %d %d",&a,&b,&c,&d);
    if(a>b){
        if(a>c){
            if(a>d){
                printf("%d is Max Number. ",a);
            }
            else {
                printf("%d is Max",d);
            }
        }
        else if(b>c){
            if(b>d){
                printf("%d is Max Number. ",b);
            }
            else {
                printf("%d is Max Number. ",d);
            }
        }
        else if(c>d){
            printf("%d is Maximum Number.",c);
        }
        else {
            printf("%d is Maximum Number.",d);
        }
    }
    return 0;
}
```



## Flow chart:



```

#include <stdio.h>
int main(){
    int i, num, product;
    for(i=1, product=1; i<=4; ++i){
        printf("Enter num%d:", i);
        scanf("%d", &num);
        if(num==0)
            continue;
        product*=num;
    }
    printf("product=%d", product);
    return 0;
}

```

## আউটপুট:

Enter num1:3  
Enter num2:0  
Enter num3:-5  
Enter num4:2  
product=-30

## ব্যাখ্যা:

উপরের প্রোগ্রামটিতে num=0 বর্ষন হবে, তখন কলিন্ট স্টেটমেন্ট product\*=num স্টেটমেন্টটি কিপ করবে এবং লুপ চলিয়ে যাবে।

**h. goto Statement:** goto স্টেটমেন্ট প্রোগ্রাম C তে ব্যবহার করা হয় সূচিত প্রোগ্রামের এক ছান থেকে অন্য ছানে যাওয়ার জন্য। goto স্টেটমেন্ট ব্যবহার করে প্রোগ্রামের স্থানিক ধারা তেস্বে প্রোগ্রামের মে঳েন জায়গার তলে যাওয়া যায়, কোন রকম বাধা goto স্টেটমেন্ট মানে না অর্থাৎ কোনো সজিক মানে না এই goto স্টেটমেন্ট।

goto স্টেটমেন্ট এর গঠন-

goto label;

//সেভেল এর দ্বারা এ ছান সুবানো হয় যেখানে প্রোগ্রামটি যাবে।

-----

-----

-----

## Label Statements

label এর ছানে যেকোনো নাম ব্যবহার করা যাবে। এখানে label হলো স্থানিক। বর্ণ প্রোগ্রামটি তার goto স্টেটমেন্টে আসবে তখন সে স্থানের label এর কাছে চলে যাবে। মাকের প্রোগ্রাম আর পড়বে না। এখানে আমরা বলতে পারি goto স্টেটমেন্ট, প্রোগ্রাম এর কমতা label-এর কাছে দিবে নিবে আমরা নিচ থেকে উপরে বা উপর থেকে নিচে যেকোনো ছানে যেতে পারবো এর দ্বারা। ইছে করলে আমরা অনেকগুলো label ব্যবহার করতে পারি একটা প্রোগ্রাম এর ভিতর। সেকেরে অবশ্যই label নাম ভিন্ন হতে হবে।

এখন আমরা একটি প্রোগ্রাম করে সেখাবো। ধৰা যাব এমন একটি প্রোগ্রাম যেখানে আমরা বাল ইনপুট দিবে এবং প্রোগ্রাম আমাকে বলে নিবে আমর ভোটার হবার বাস হয়েছে কিনা। এখানে একটু বলে রাখি, একটা প্রোগ্রাম আমরা অনেকভাবে করতে পারি, অনেক সজিক ব্যবহার করে। এই প্রোগ্রামটা আমরা goto স্টেটমেন্ট দিয়ে করবো।

```

#include <stdio.h>
#include <conio.h>

```

```

int main(){
    int age;
    printf("Enter your age:");
    scanf("%d", &age);
    if(age >= 18)
        goto Vote;
    else
        goto NoVote;
    NoVote:
    printf("you are not eligible to
vote");
    goto end;
    Vote:
    printf("you are eligible for
voting");
    goto end;
    end:
    return 0;
}

```

উপরের প্রোগ্রামটিতে num=0 বর্ষন হবে, তখন কলিন্ট স্টেটমেন্ট product\*=num স্টেটমেন্টটি কিপ করবে এবং লুপ চলিয়ে যাবে।

Enter you age : 25

You are eligible for voting

## ডি পি প্রোগ্রাম লুপ (loop) ডি

## লুপ(loop) [NTRCA-2010, 19]:

লুপ(loop) এর মাধ্যমে কোন প্রোগ্রাম এর নিদিষ্ট কোন অংশ বারবার এক্সিকিউট করা যায়, যতক্ষণ পর্যন্ত তাকে দেয়া শর্তটি মিথ্যা না হয়।

অশ. ১. প্রোগ্রাম সি তে কয় ধরনের লুপ(loop) রয়েছে?

উত্তর : ৩ ধরনের যথা-

a. while loop

b. do...while

c. for loop

a. **while loop :** হোয়াইল লুপ এর কাজ হল একটা প্রোগ্রামের টেস্ট এক্সেশন চেক করে যে এক্সেশনটি সত্য নাকি মিথ্যা। যদি টেস্ট এক্সেশনটি সত্য হয় তাহলে এটা হোয়াইল লুপের বীর্তির ভেতরের কোড তলে এক্সিকিউট করবে, মানে ব্রাকেট এর ভেতরের কোড তলে এক্সিকিউট করবে। তারপর এটি আবার টেস্ট এক্সেশনটি চেক করবে যে এটি সত্য নাকি মিথ্যা। এই প্রক্রিয়া টি ততক্ষণ তারে টেস্ট এক্সেশনটি যতক্ষণ পর্যন্ত মিথ্যা না হবে।

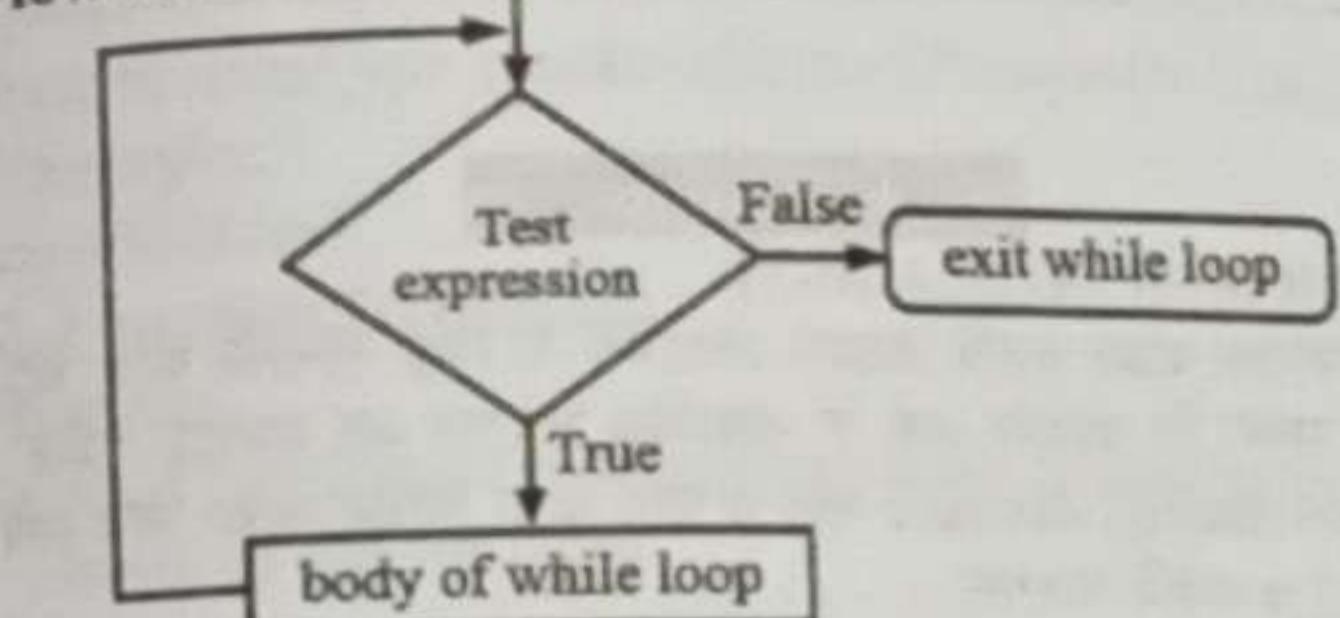
## Syntax:

```

while (condition){
    //statements to be executed.
}

```

## Flow chart:



হোয়াইল লুপের উদাহরণট এমন একটি সি প্রোগ্রাম লিখতে হবে যেটা যে কোন একটি নামাবের Factorial value বের করবে এবং নামাবটি ইটজার ইনপুট করবে।  
(ইলিস্ট n এর ফ্যাক্টোরিয়াল=n\*(n-1)\*.....4\*3\*2\*1)  
[হোয়াইল লুপ কিভাবে কাজ করে সেটা দেখানোর জন্য একটি সি প্রোগ্রাম দেওয়া হল।]

```

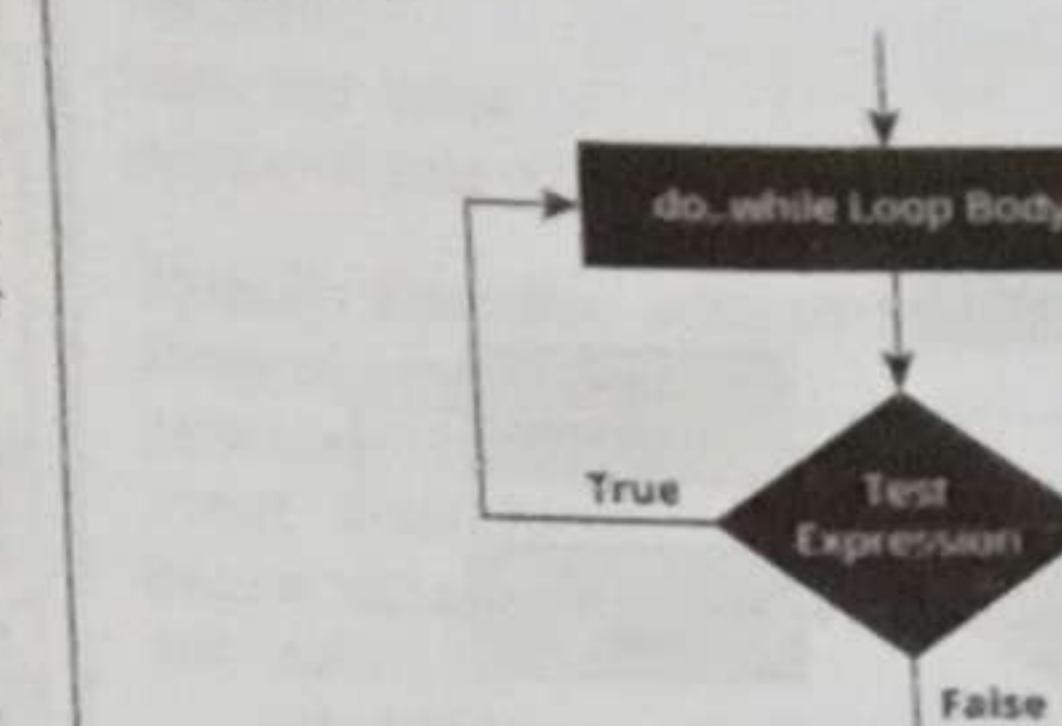
#include <stdio.h>
int main() {
    int number,factorial=1;
    printf("Enter a number.\n");
    scanf("%d",&number);
    while (number>0){
        factorial=factorial*number;
        --number;
    }
    printf("Factorial=%d",factorial);
    return 0;
}

```

## আউটপুট:

Enter a number: 5  
Factorial=120

## Flow chart:



থথমে বীর্তির ভেতরের do লুপ এক্সিকিউট হবে তারপর প্রোগ্রাম এক্সেশন চেক করবে। এক্সেশন যদি সত্য হয় তাহলে do আবার এক্সিকিউট হবে এবং এক্সেশন মিথ্যা না হওয়া পর্যন্ত লুপ চলতে থাকবে।

```

#include <stdio.h>
int main(){
    int sum=0,num;
    do {
        printf("Enter a number: ");
        scanf("%d",&num);
        sum+=num;
    }
    while(num!=0);
    printf("sum=%d",sum);
    return 0;
}

```

## আউটপুট:

Enter a number: 3  
Enter a number: -2  
Enter a number: 0  
sum=1

**c. for loop:** initialization স্টেটমেন্ট থধু একবার এক্সিকিউট হয় ফর লুপ এর ভরতে। এরপর টেস্ট এক্সেশন চেক করা হয়। যদি টেস্ট এক্সেশন ভুল হয়, তাহলে লুপ ওখানেই শেষ হয় আর যদি টেস্ট এক্সেশন সঠিক হয়, তাহলে তা block এর ভেতর চুক যা আছে তা এক্সিকিউট করে। পরে প্রোগ্রাম আবার অপডেট এক্সেশন চেক করে এবং টেস্ট এক্সেশন মিথ্যা ন হওয়া পর্যন্ত কাজ চলিয়ে যাব।

[NTRCA-2014, 17]

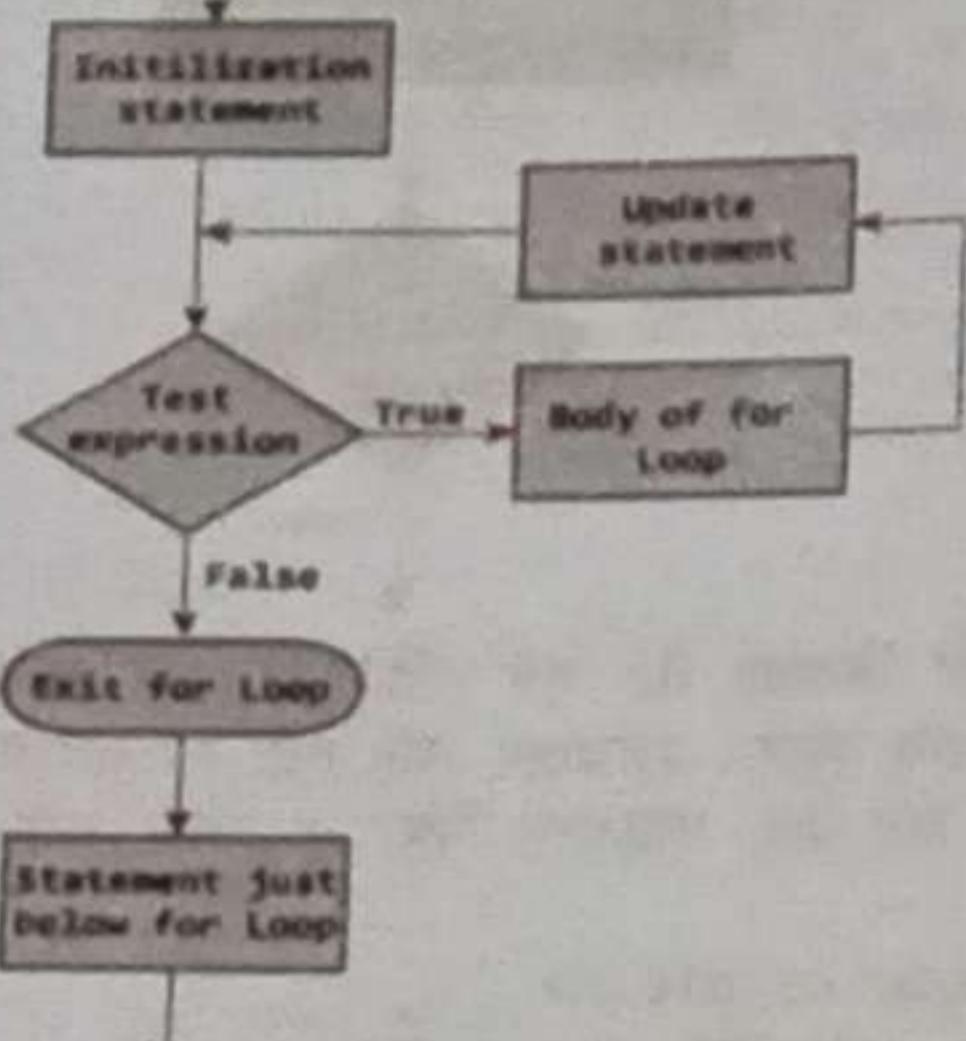
**Flow chart:**

Figure: Flowchart of for Loop

**Syntax:**

```
for(initialization ; condition; update state){
    // code's to be executed;
}
```

■ ফর লুপ এর উদাহরণ:  $1 + 2 + \dots + n$  series

**summation** [NTRCA-2010(101.....1001)]

```
#include <stdio.h>
int main(){
    int n, count, sum=0;
    printf("Enter the value of n.\n");
    scanf("%d",&n);
    for(count=1;count<=n;++count) {
        sum+=count;
    }
    printf("Sum=%d",sum);
    return 0;
}
```

**আউটপুট:**

Enter the value of n 19

Sum=190

এই প্রোগ্রামে ইউজার এর কাছে প্রথমে জানতে চাবার হয়, সে ক্ষেত্রে সংখ্যার যোগফল বের করতে চায়। ধরি, আমরা এখনে প্রথম 19 টি সংখ্যার যোগফল বের করতে চাই। এখনে count variable এর মান 1 দিয়ে শুরু করা হয়েছে, এর ফলে দেখা যাচ্ছে যে টেস্ট এক্সপ্রেশন টি সত্য। তাই তা ব্রক এর মধ্যে থাকা এক্সপ্রেশন টি এক্সেকিউট করবে। পরে  $++\text{count}$  এক্সপ্রেশন এর মাধ্যমে count এর মান এক বাড়বে।

তখনও টেস্ট এক্সপ্রেশন টি সত্য থাকে। এর ফলে প্রোগ্রাম টি আবার এক্সেকিউট হবে। অভাবে শূণ্যটি চলতে থাকবে। যখন count এর মান 20 হয়ে যাবে, তখন টেস্ট এক্সপ্রেশনটি মিথ্যা হবে এবং তাতে করে শূণ্যটি বক হয়ে যাবে।

**প্রশ্ন 2. Difference Between While and do While loop.[ICT-2014]**

While	do While
Syntax: while(condition){ }	Syntax: do{ }while(condition);
loop শেষে সেমিকোলন (;) ব্যবহার হয় না। প্রথমেই শর্ত চেক করা হয়। While loop টি এন্ট্রি controlled loop.	loop শেষে সেমিকোলন (;) ব্যবহার হয়। শেষে শর্ত চেক করা হয়। Do while loop টি এন্ট্রি controlled loop.

**॥ Functions ॥****প্রশ্ন 1. ফাংশন কি? [NTRCA-2015]**

উভয় ফাংশন হচ্ছে একটি প্রোগ্রাম সেগমেন্ট যা কিছু সুনির্দিষ্ট কাজ করে থাকে। সকল সি প্রোগ্রাম এক বা একাধিক ফাংশন এর সমন্বয়ে গঠিত। আমরা যে main() declare করি ও তার মধ্যে বিভিন্ন কাজ করি সেই main() ও একটি ফাংশন।

**প্রশ্ন 2. ফাংশন ব্যবহার সুবিধা সমূহ লিখ।**

উভয় ফাংশন ব্যবহার সুবিধা সমূহ-

- Function provides modularity।
- ফাংশন পূর্ণব্যবহারযোগ্য কোড।
- বড় প্রোগ্রাম, debugging এবং editing করা সহজ।
- বিভিন্ন মডিউলে এ ভাগ করা যায়।
- Function কে লাইব্রেরি তে সংযুক্ত করে প্রযোজন এর সময় ব্যবহার করা যায়।
- প্রোগ্রাম এর আকার ছোট হয়।

**প্রশ্ন 3. C তে কয় ধরনের ফাংশন রয়েছে?**

উভয় C তে দুই ধরনের ফাংশন রয়েছে।

**Built In Function :** C তে by default যেসব ফাংশন আগেই তৈরি করা থাকে সেগুলোকেই আমরা বিস্তৃত ইন ফাংশন বলি। এগুলো আমাদের জন্য আগেই তৈরি করা থাকে, আমাদের কাজ তখন করা e.g. scanf(), printf(), strcpy(), strlwr(), strcmp(), strlen(), strcat(), etc.

**User Defined Function** আমাদের নানা কাজে নানা ধরণের ফাংশন প্রযোজন হয়, এই সব ক্ষেত্রে আমরা নিজেরাই ফাংশন তৈরি করে নিতে পারি বা অন্যের তৈরি করা ফাংশন ব্যবহার করতে পারি। যে সব ফাংশন ইউজার অর্ধাং আমাদের (আমার নিজের বা অন্য কোন তেলেগুগৱের) তৈরি করা সেগুলোকে ইউজার ডিফাইনড ফাংশন বলা হয়।

একটি ফাংশনের সাধারণ ফরম্যাট হল:

```
Return_DataType function_name (parameter list){
// Body
}
```

প্রশ্ন 8. ফাংশনের বিভিন্ন অংশগুলো ব্যাখ্যা কর।  
উভয় ফাংশনের বিভিন্ন অংশগুলো ব্যাখ্যা।

• **Return\_Data\_Type:** ফাংশনটি কি ধরণের data type এর value return করবে operation শেষে তা উল্লেখ করতে হবে।

• **ইনপুট বা parameter:** function এর parameter একাধিক হলে কমা দিয়ে data type সহ declare করা হয়।

• **Prototype:** যেখানের প্রোটোটাইপ মেথড ডেফিনিশন এর সাথে সম্পর্কিত। কোনো মেথড এর প্রোটোটাইপ বলতে তার সঠিক নাম, রিটার্ন টাইপ, আর্গুমেন্ট তালিকা, নির্ধারিত ফরম্যাটে বোধায়। [NTRCA-2017]

**Output:**

Enter the radius : 5  
Area of Circle =78.500000

Type 2 : Function with no arguments but return.  
Return(Area of Square).

```
#include<stdio.h>
float area(); // Prototype
Declaration
int main(){
    printf("Area of Square = %f",area());
    return 0;
}
```

**1. Function Prototype****Syntax:**

Return dataType functionName (Parameter List);

**Example:**

int addition();

**2. Function Definition****Syntax:**

returnType functionName(Function parameters){

//body of the function

}

**॥ Category of Function Calling ॥**

[NTRCA-2015]

Function Type	Parameter	Return Value
Type 1	Not Accepting Parameter	Not Returning Value
Type 2	Not Accepting Parameter	Returning Value
Type 3	Accepting Parameter	Not Returning Value
Type 4	Accepting Parameter	Returning Value

**Type 1 : Function with no arguments and no Return (Area of Circle Program).**

```
#include<stdio.h>
void area(); // Prototype
```

```
int main(){
    area();
}
```

```
void area(){
    float area_circle;
```

```
    float rad;
    printf("\nEnter the radius : ");
    scanf("%f",&rad);
    area_circle = 3.14 * rad * rad ;
    printf("Area of Circle = %f",area_circle);
    return 0;
}
```

Output:  
Enter the radius : 5  
Area of Circle =78.500000

**Output:**

Enter the length and breadth : 10 12

Area of Rectangle = 120.000000

**Type 3 : Function with arguments but no return.**

Return(Area of Rectangle).

```
#include<stdio.h>
#include<conio.h>
void area(float length,float breadth);
int main(){
    float breadth,length;
    printf("Enter the length and breadth : ");
    scanf("%f%f",&length,&breadth);
    area(length,breadth);
    return 0;
}
```

```
void area(float length,float breadth){
    float ar;
    ar = length * breadth;
    printf("Area of Rectangle = %f",ar);
}
```

**Output:**

Enter the length and breadth : 10 12

Area of Rectangle = 120.000000

**Type 4 : Function with arguments and Return(Adding two number).**

```
#include<stdio.h>
int sum(int , int); //Prototype
```

```
int main(){
    int x,y,result;
    printf("Enter value of x and y: ");
    scanf("%d %d", &x, &y);
}
```

```
result = sum(x,y);
printf("Sum = %d",result);
}
```

## Structure Programming (C Program)

```

result = sum(x, y); // calling of function-sum
printf("Sum :%d", result);
return 0;
}
int sum(int a, int b) { //function definition
    int res;
    res = a + b;
    return res ;
}
Output:
Enter value of x and y: 10 12
Sum :12

```

**প্রশ্ন ৫.** Call by value এবং Call by reference কলে কি বুঝতে অবধা, C তে function Calling এর পদ্ধতি অবধা, Call by value কি ? অবধা, Call by reference কি ? [NTRCA-2014, Multiple Ministry-2016]

Type	Description
Call by Value	Call by Value টত argument এর actual value parameter হিসেবে পাঠানো হয়, তাই original value modified হয় না।
Call by Reference	Call by Reference এ argument এর address parameter হিসেবে পাঠানো হয়, তাই original value modified হয়।

[Type:1,2,3,4 call by value এর example]

### Example of Call by Reference:

```

#include <stdio.h>
void increment(int *var){
    *var = *var+1;
}
int main(){
    int num=20;
    increment(&num);
    printf("Value of num is: %d", num);
    return 0;
}
Output:
Value of num is: 21

```

**প্রশ্ন ৬.** C program for prime numbers between 1 to N (Prime series) using function . [Ministry of Edu-2016], NTRCA-2015, 16]

```

#include<stdio.h>
int prime(int N){
    int i, j, P;
    printf("The Prime numbers from 1 to N:");
    for(i = 2; i <= N; i++){
        P = 0;
        for(j = 2; j <= i/2; j++){

```

```

            if(i % j == 0){
                P= 1;
                break;
            }
        }
        if(P==0)
            printf("%d ",i);
    }
}
int main(){
    int Num;
    printf("Enter the value of N: ");
    scanf("%d",&Num);
    prime(Num);
    return 0;
}
Output:
Enter the value of N:10
The Prime numbers from 1 to N: 2 3 5 7

```

### ॥ রিকুর্সিভ ফাংশন ॥

**প্রশ্ন ১.** রিকুর্সিভ ফাংশন কি ? [NTRCA-2016] [ICT D-AP-2021]  
উত্তর : কোন ফাংশন যখন নিজেকে কল করে তখন তাকে রিকুর্সিভ ফাংশন বলা হবে থাকে।

[NB : Recursive function এর উদাহরণ প্রয়োগ অংশে ব্যাখ্যাদহ দেওয়া আছে।]

**প্রশ্ন ২.** রিকুর্সিভ ফাংশনের সুবিধা কী?

উত্তর : রিকুর্সিভ ফাংশনের সুবিধা নিম্নরূপ :

- প্রয়োগের জটিলতা হ্রাস পাও।
- কম সংখ্যক Variable এর প্রয়োজন হয়।
- প্রয়োগ সহজ ও সুব্লিম হয়।

### ■ Difference Between Recursion and Iteration

[NTRCA 2010]

Recursion এবং Iteration উভয়ই set of instructions বাইবার একে করে।

RECURSION	ITERATION
রিকুর্সিভ ফাংশনে, কেবল সমাপ্তির শর্ত (বেলে কেস) নির্দিষ্ট করা হয়।	Iteration ফাংশনে, তরুণ, শর্ত, execution statements এবং মান পরিবর্তন (বৃক্ষি এবং হ্রাস) অন্তর্ভুক্ত করা হয়।
Recursion কল কর্মকর না করে ফাংশনটি ফিরে আসতে বাধ্য করার জন্য একটি শর্ত বিবৃতি ফাংশনের body টতে অন্তর্ভুক্ত করা হয়।	একটি নির্দিষ্ট শর্তে না পৌছানো পর্যন্ত পুনরাবৃত্তি বিবৃতি বাইবার কার্যকর করা হয়।
Recursion সর্বনা ফাংশনে প্রয়োগ করা হয়।	Iteration সাধারণত iteration statements বা "loops" এ প্রয়োগ করা হয়।

### ॥ Array ॥

**প্রশ্ন ১.** Array কি ? এর বৈশিষ্ট্য লিখ। [NTRCA-2015]  
উত্তর : Array (অ্যারে) হল একই data type এর collection of element (একটি ডাটা গঠন পদ্ধতি) যা উপাদান বা element কে গ্রুপ আকারে ধারণ করতে পারে। তবে সেই উপাদান সমূহকে একই টাইপের ডাটা টাইপ হতে হবে। যেমনঃ পূর্ণ সংখ্যা বা স্ট্রিং।

বৈশিষ্ট্য :

- Array size constant হতে হবে।
- element সংরক্ষণ করার জন্য Contiguous (adjacent) memory allocation করা হয়।
- যে কোন ডাটা টাইপ হতে পারে।

**প্রশ্ন ২.** একমাত্রিক অ্যারে কাকে বলে ?

উত্তরঃ যে অ্যারেতে একটি মাত্র কলাম ও একাধিক সারি অবধা একটি মাত্র সারি এবং একাধিক কলাম উপস্থাপন করা হয় তাকে একমাত্রিক অ্যারে বলা হয়।

**প্রশ্ন ৩.** দ্বিমাত্রিক অ্যারে কাকে বলে ?

উত্তরঃ যে অ্যারেতে একাধিক সারি ও একাধিক কলামে ডেটা উপস্থাপন করা হয় তাকে দ্বিমাত্রিক অ্যারে বলা হয়।

## Structure Programming (C Program)

Stack ব্যবহার হয়।	Stack ব্যবহার হয় না।
দ্বিমাত্রিক সম্পর্ক।	দ্রুতগতি সম্পর্ক।
Recursion কোডের আকার হ্রাস করে।	Iteration কোডের আকার দীর্ঘায়িত করে।

**■ Library Function:** Library function হল Language এ দেয়া default function's. Library function এ সকল ফাংশন এর corresponding header file আছে।

### কিছু জনপ্রিয় লাইব্রেরী ফাংশন:

Function name	Header file	কাজ
scanf()	stdio.h	Input দেয়া
printf()	stdio.h	Output দেয়া
pow(x,y)	math.h	Return $x^y$
sqrt(d)	math.h	Return d(square root of d)
isalnum(c)	ctype.h	Return true if character c is either A-Z, a-z, 0-9
isalpha(c)	ctype.h	Return true if character c is either A-Z, a-z
isdigit(c)	ctype.h	Return true if character c is either 0-9
isupper(c)	ctype.h	Return true if character c is either A-Z
islower(c)	ctype.h	Return true if character c is either a-z
abs()	stdlib.h	Return the absolute value

### ■ One Dimensional Array বোঝনার নিয়ম:

Data type arr\_name[array size];  
Example: int arr[7];

arr[0] arr[1] arr[2] arr[3] arr[4] arr[5] arr[6]

### ■ Array initialization বোঝনার নিয়ম :

data\_type arr\_name [arr\_size]={value1, value2, value3,...};

### Example:

int age [5] = {0, 1, 2, 3, 4};

### Example program for one dimensional array in c:

```

#include<stdio.h>
int main (){
    int i;
    int Arr [5] = {10,20,30,40,50};
    Arr[5]={0};
    Arr [0] = 10;
    Arr [1] = 20;
    Arr [2] = 30;
    Arr [3] = 40;
    Arr [4] = 50;
    for(i=0;i<5;i++){
        printf("value of Arr[%d] is %d \n", i, Arr[i]);
    }
}

```

### OUTPUT:

value of Arr [0] is 10  
value of Arr [1] is 20  
value of Arr [2] is 30  
value of Arr [3] is 40  
value of Arr [4] is 50

### ■ Two-dimensional array syntax:

Data\_Type arr\_Name [num\_Of\_Rows][num\_of\_column];

Example: int arr [2][2];

0,0	0,1
1,0	1,1

■ 2D Array initialization syntax:

```
data_type arr_name [2][2] = {{0,0},{0,1},{1,0},{1,1}};
```

**Q String**

String: String হচ্ছে ক্যারেক্টর টাইপের আবেদন। যেটি শেষ হয় null('0') ক্যারেক্টর দিয়ে।

**Syntax:**

```
char variable_name[size]; or  
string variable name[size];
```

**Example For C String:**

1. char string [20] = {'f', 'r', 'e', 's', 'h', '2', 'r', 'e', 'f', 'r', 'e', 's', 'h', '0'};
2. char string [20] = "bitBoxGuide";
3. char string [] = "bitboxbd.com";

**Example program for c string:**

```
#include <stdio.h>  
int main ()  
{  
    char string [20] = "bitBox ";  
    printf ("The string is : %s \n", string );  
    return 0;  
}
```

**OUTPUT:**

The string is: bitBox

সি স্ট্রিং পরিচালনার ফাঁশন: সি প্রোগ্রাম এ অনেক ধরণের স্ট্রিং অপারেশন রয়েছে যেতেলো আপনি মানুয়ালি(manually) সম্পাদন করতে পারেন। যেমন- স্ট্রিং এর দৈর্ঘ্য(length) নির্ণয়, স্ট্রিং কপি(copy) করা, দুটি স্ট্রিং এর সংযোগ(concatenation) ইত্যাদি।

কিন্তু প্রোগ্রামারদেরকে সহজ করে দেওয়ার জন্য এ ধরনের অনেক লাইব্রেরী ফাঁশন <string.h> হেভেড ফাইলের অধীনে ইতোমধ্যেই ডিফাইনড(define) করা আছে।

লাইব্রেরী ফাঁশন ব্যবহার করে স্ট্রিং পরিচালনা: এই অধ্যায়ে লাইব্রেরী ফাঁশন যেমন- gets(), puts(), strlen() ইত্যাদি ফাঁশন ব্যবহার করে সি প্রোগ্রাম এ স্ট্রিং পরিচালনা সম্পাদন করতে পারেন। যেমন- স্ট্রিং এর দৈর্ঘ্য(length) নির্ণয়, স্ট্রিং কপি(copy) করা, দুটি স্ট্রিং এর সংযোগ(concatenation) ইত্যাদি। এছাড়া ইউজার থেকে স্ট্রিং(string) এবং উপর বিভিন্ন অপারেশন চালাতে পিষ্টেন। লাইব্রেরী ফাঁশন- স্ট্রিং পরিচালনা সমস্যার ধরন অনুযায়ী আয়োজিত করা হচ্ছে। যদি মানুয়ালি নির্ণৃতভাবে স্ট্রিং পরিচালনা(manipulation) করতে চান তাহলে ইহা অনেক জটিল হতে পারে এবং বেশীরভাবে সমস্যার অভিজ্ঞতা আপনাকে সমস্যায় পড়তে হতে পারে। তাই এই সমস্যা থেকে উত্তরান্তের জন্য সি লাইব্রেরী ফাঁশন ব্যবহারের জন্য সুপারিশ করা হচ্ছে।

স্ট্রাকচার লাইব্রেরী ফাঁশন: "string.h" এর অধীনে স্ট্রিং পরিচালনার জন্য অস্বীকৃত ফাঁশন রয়েছে। স্ট্রিং পরিচালনার জন্য সচরাচর ব্যবহৃত ফাঁশন এবং তাদের কাজ নিচে তুলে ধরা হলো:

**C String Functions:** String.h header file supports all the string functions in C language.

String functions	Description
strcat()	দুটি string কে সংযুক্ত করার জন্য ব্যবহার করা হয়।
strcpy()	একটি string আর একটি string এ কপি করার জন্য ব্যবহার করা হয়।
strlen()	string এর length বের করার জন্য ব্যবহার করা হয়।
strcmp()	দুটি string কে তুলনা করার জন্য ব্যবহার করা হয়। getchar() function reads character from keyboard.
putchar()	putchar() function writes a character to screen.
gets()	gets() function reads string from user
puts()	puts() function prints the string
strrstr()	Returns pointer to last occurrence of str2 in str1.
strlwr()	স্ট্রিংকে ছোট হাতের বর্ণে(lowercase) রূপান্তর করে।
strupr()	স্ট্রিংকে বড় হাতের বর্ণে(uppercase) রূপান্তর করে।
strrev()	স্ট্রিংকে বিপরীত(reverse) রূপে সাজায়।
strstr()	স্ট্রিং থেকে উপ স্ট্রিং খুঁজে বের করে।

**Simple input and output program using string.**

```
#include<stdio.h>  
int main(){  
    char str[20];  
    gets(str);  
    puts(str);  
    return 0;  
}
```

pointer ব্যবহার করে pointer ব্যবহার করেও স্ট্রিংকে নিম্নে ন্যায় ইনিশিয়ালাইজ(initialized) করা যেতে পারেং:

```
char *c = "abcd";
```

ইউজার থেকে স্ট্রিং read করা: অন্য সকল ভাষা টাইপের মতই scanf() ফাঁশন ব্যবহার করে আপনি স্ট্রিং read করতে পারেন। যাইহেকে scanf() ফাঁশন প্রবেশকৃত শব্দ-সমূহের তথ্যাবলী প্রথম শব্দকে গ্রহণ করে। white space বা তথ্যাবলী স্পেসের দেখা পাওয়া যাবে এই ফাঁশনের সমাপ্তি ঘটে।

ইউজার থেকে input গ্রহণ:

```
char c[20];  
scanf("%s", c);
```

উদাহরণঃ scanf() ফাঁশন ব্যবহার করে স্ট্রিং read করা

```
#include <stdio.h>  
int main(){  
    char name[20];  
    printf("Enter your name: ");  
    scanf("%s", name);  
    printf("Your name is %s.", name);  
    return 0;  
}
```

gets(name); // ইউজার থেকে স্ট্রিং পাঠ করার জন্য ফাঁশন  
printf("Name: ");  
puts(name); // স্ট্রিং প্রদর্শনির ফাঁশন  
return 0;

আউটপুট:  
Enter name: bitBox IT  
Name: bitBox IT

ফাঁশনের মধ্য দিয়ে স্ট্রিং Pass: ফাঁশনের মধ্য দিয়ে স্ট্রিং Pass করানো সি প্রোগ্রাম এ স্ট্রিং char array এর মতই। সূতরাং এদেরকে array এর মত একটি পক্ষতাতে ফাঁশনের মধ্য দিয়ে pass করানো যেতে পারে।

```
#include <stdio.h>  
void displayString(char str[]);  
int main(){  
    char str[50];  
    printf("Enter string: ");  
    gets(str);  
    displayString(str); // ফাঁশনের মধ্য দিয়ে স্ট্রিং কে  
    // অতিক্রম করানো হয়।  
    return 0;  
}
```

উদাহরণঃ getchar() ফাঁশন ব্যবহার করে টেক্সট(text) এর একটি স্ট্রিংকে পাঠ করা হয়।  
#include <stdio.h>  
int main(){  
 char name[30], ch;  
 int i = 0;  
 printf("Enter name: ");  
 while(ch != '\n'){ // ইউজার enter বাটনে ক্লিক করলেই সমাপ্তি  
 ch = getchar();  
 name[i] = ch;  
 i++;  
 }  
 name[i] = '\0'; // সবশেষে null ক্যারেক্টর যুক্ত করা  
 printf("Name: %s", name);  
 return 0;  
}

উদাহরণঃ প্রোগ্রাম স্ট্রিং(string) s কে main() ফাঁশন থেকে ইউজার ডিফাইনড(user-defined) ফাঁশন displayString() এর মধ্য দিয়ে pass করানো হয়েছে। ফাঁশন ডিক্রিয়ারেশনে str[] হলো ফর্মাল অ্যার্মেন্ট(formal argument)।

সি strstr() ফাঁশন: Strstr() ফাঁশন প্রদত্ত স্ট্রিং এর মধ্যে মিলিত স্ট্রিং এর প্রথম সংযুক্তের প্রিটার্ন করে। এটি substr এর মিলিত প্রথম অক্ষর থেকে সর্বশেষ অক্ষর পর্যন্ত প্রিটার্ন করে।

সিনট্যাক্স:

```
char *strstr (const char *string, const char *match)
```

সিনট্যাক্সের ব্যাখ্যা:  
string: ইহা দ্বারা স্ট্রিং এর সম্পূর্ণ অংশকে বুকায়। যেখান থেকে substring সার্চ করা হবে।  
match: ইহা হচ্ছে submatch যা সম্পূর্ণ স্ট্রিং থেকে সার্চ করা হবে।

```
#include<stdio.h>  
#include<conio.h>  
#include<string.h>  
int main(){  
    char str[100] = "I love programming with C";  
    char *sub;  
    clrscr();  
    sub = strstr(str, "programming");  
    printf("\nSubstring is: %s", sub);
```

## Structure Programming (C Program)

```

getch();
}

আউটপুট:
Substring is: programming with C

```

### ৪. পয়েন্টার (Pointer) :

**পয়েন্টার (pointer) [NTRCA-2012]**  
পয়েন্টার হচ্ছে মেমোরি অ্যাড্রেস এর রেফারেন্স। একদিকে মেমন, তেরিয়েবল হচ্ছে ভ্যালু রাখার কন্টেইনার অন্যদিকে পয়েন্টার হচ্ছে মেমোরিতে ভ্যালুটি মেমোরির দেখানে জন্ম আছে সেখানকার অ্যাড্রেস এর রেফারেন্সের ধারক।

সি প্রোগ্রামিং পয়েন্টার: সি এবং সি++ প্রোগ্রামিং এর শক্তিশালী তৈরিত হলো পয়েন্টার (Pointer) যার মাধ্যমে ইহা অন্যান্য জনপ্রিয় প্রোগ্রামিং ল্যাঙ্গুজেজে মেমন- পাইন- এবং জাভা থেকে বেশী প্রাথমিক পার। মেমোরি এক্সেস(access) করার জন্ম এবং মেমোরি এক্সেস নিম্নভাবে পরিচালনা করার জন্য সি প্রোগ্রামিং এ পয়েন্টার ব্যবহৃত হয়।

**প্রশ্ন . Pointer এর বৈশিষ্ট্য লিখ।**

**উত্তর: Pointer এর বৈশিষ্ট্য-**

- i. Pointer variable declare করার সময় (\*) চিহ্ন পয়েন্টার variable এর নামের সাথে ব্যবহার করা হলেও এটি নামের অর্থুর্ক নয়।
- ii. কোন pointer variable কে যে data type এ declare করা হয় সে তথ্যটি এই type এর data point করতে পারে।
- iii. Declare এর পূর্বে মান নির্ধারণ করতে হয়।
- iv. সাধারণ variable এর মত pointer variable declare করা যায়।

সি মেমোরি এক্সেস: পয়েন্টারের ধারণা নেওয়ার পূর্বে চূলুন সি প্রোগ্রামিং এক্সেসের সাথে পরিচিত হয়ে নিই। প্রোগ্রামে যদি count একটি ভ্যারিয়েবল থাকে তাহলে &count আপনাকে মেমোরির এক্সেস দিবে। এখানে & কে রেফারেন্স অপারেটর বলা হয়।

যখন scanf() ফাংশন ব্যবহার করবেন তখন অবশ্যই এই প্রতীকটি দেখবেন। ইউনিয়ন কর্তৃক ইনপুট ভ্যালু ভ্যারিয়েবলের এক্সেস করার জন্য ইহা ব্যবহৃত হয়।

```

scanf("%d",&count);
/* সি প্রোগ্রামিং এ রেফারেন্স অপারেটর ভালভাবে বুঝার জন্য এই
প্রোগ্রাম : */
#include <stdio.h>
int main(){
    int count=5;
    printf("Value: %d\n",count);
    printf("Address: %u",&count); // লক্ষ্য করলে দেখবেন
    যে, count ভ্যারিয়েবলের পূর্বে ampersand(&) ব্যবহার করা
    হয়েছে।
    return 0;
}

```

### আউটপুট:

Value: 5  
Address: 2686778  
লেটেষ্ট এই কোড ব্যবহার করে আপনি এক্সেসের বিভিন্ন ভ্যালু পেতে পারেন।

উপরের সোর্স কোডে ভ্যালু 5 মেমোরির 2686778 লোকেশনে জন্ম হয়। count তথ্যটি সেই লোকেশন(location) এর নাম।

সি Pointer ভ্যারিয়েবল সি প্রোগ্রামিং এ এক ধরণের বিশেষ ভ্যারিয়েবল যার মাধ্যমে অন্যান্য ভ্যারিয়েবলের এক্সেস জন্ম আসে। ইহাকে পয়েন্টার ভ্যারিয়েবল বা তথ্য পয়েন্টার বলা হয়।

**Pointer ডিক্রিমেশন:**

```

data_type* pointer_variable_name;
int* p;
উপরের স্টেটমেন্টে p কে int টাইপের পয়েন্টার ভ্যারিয়েবল হিসাবে
ডিক্রিমেশন(declare) করা হয়েছে।

```

**Example: #include <stdio.h>**

```

int main(){
    int x = 10;
    int *num = &x;
    printf("Address of variable x is:%d\n", &x);
    printf("Value of variable num is:%d", num);
    printf("\nAddress of variable num is: %p", &num);
    return 0;
}

```

### Output:

```

Address of variable x is:2686748
Value of variable num is: 10
Address of variable num is: 0x7fdc58

```

**Pointer এর ক্ষেত্রে এক যে data type এ declare করা হয় সে তথ্যটি এই type এর variable কে Point করতে পারে।**

int a, b, c;	int * S;
int * s;	float a, b;
S = &a;	S = &a;
<b>Right statement</b>	<b>Wrong statement</b>

সি রেফারেন্স(&) এবং ডিরেফারেন্স(\*) অপারেটর:

ইতোপূর্বে বর্ণনা করা হয়েছে যে, & কে রেফারেন্স(reference) অপারেটর বলা হয়। ইহার মাধ্যমে আপনি ভ্যারিয়েবলের এক্সেস পাবেন। অনুকূলভাবে, অন্য একটি অপারেটর রয়েছে যার মাধ্যমে আপনি এক্সেস থেকে ভ্যালু পেতে পারেন। ইহাকে ডিরেফারেন্স(\*) অপারেটর বলা হয়।

নিচের উদাহরণে পয়েন্টার, রেফারেন্স(reference) অপারেটর এবং ডিরেফারেন্স(dereference) অপারেটরের ব্যবহার স্পষ্টভাবে ব্যাখ্যা করা হয়েছে।

নোটঃ পয়েন্টার ডিক্রিমেশন(declaration) এ ব্যবহৃত \* চিহ্ন ডিরেফারেন্স অপারেটর নয়। ইহা পয়েন্টার নির্দেশের জন্য তথ্যটি একটি প্রতীক।

## Structure Programming (C Program)

উদাহরণট পয়েন্টারের ব্যবহার শেখাব জন্য সি প্রোগ্রাম

```

#include <stdio.h>
int main(){
    int* pc;
    int c;
    c=22;
    printf("Address of c:%u\n",&c);
    printf("Value of c:%d\n\n",c);
    pc=&c;
    printf("Address of pointer pc:%u\n", pc);
    printf("Content of pointer pc:%d\n\n",*pc);
    c=11;
    printf("Address of pointer pc:%u\n", pc);
    printf("Content of pointer pc:%d\n\n",*pc);
    *pc=2;
    printf("Address of c:%u\n",&c);
    printf("Value of c:%d\n\n",c);
    return 0;
}

```

### আউটপুট:

```

Address of c: 2686784
Value of c: 22

```

```

Address of pointer pc: 2686784
Content of pointer pc: 22

```

```

Address of pointer pc: 2686784
Content of pointer pc: 11

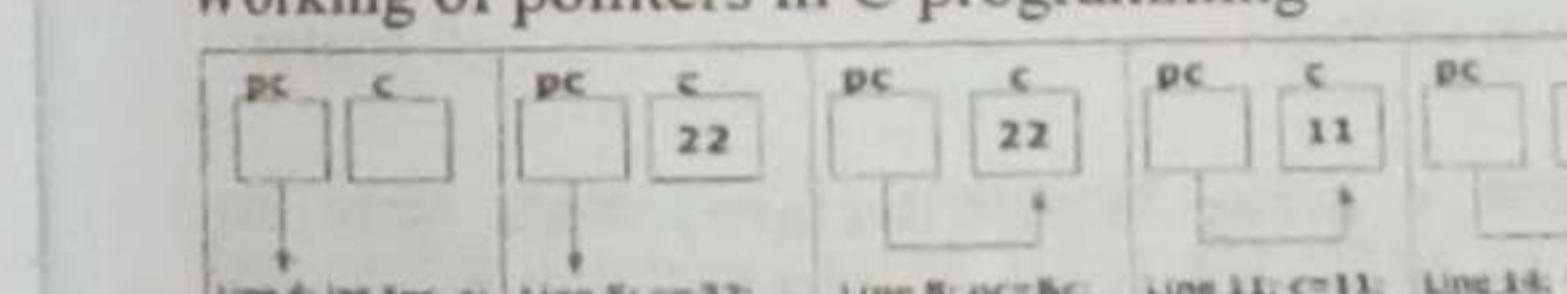
```

```

Address of c: 2686784
Value of c: 2

```

### Working of pointers in C programming



উপরের প্রোগ্রাম এবং চিত্রের ব্যাখ্যা:

(i). int\* pc; পয়েন্টার ভ্যারিয়েবল pc তৈরি করে এবং int c; সাধারণ ভ্যারিয়েবল c তৈরি করে।

যেহেতু pc এবং c উভয়কেই ইনিশিয়ালাইজ(initialized) করা হয়নি, সুতরাং পয়েন্টার pc এক্সেস বা র্যান্ডম(random) এক্সেস কোনো ক্ষেত্রেই নির্দেশ করে না। একইভাবে ভ্যারিয়েবল c এর এক্সেস এসাইন(assign) করা হয়েছে কিন্তু ইহা random/garbage ভ্যালু ধারণ করে।

(ii). c=22; ভ্যারিয়েবল c এ 22 এসাইন(assign) করে অর্থাৎ 22 ভ্যারিয়েবল p এর মেমোরি লোকেশনে জন্ম আসে।

মনে রাখবেন, c(ভ্যারিয়েবল) এর এক্সেস(&c) প্রিন্ট করার সময় আমরা %d এর পরিবর্তে %u ব্যবহার করবো যেহেতু এক্সেস সবসময় অনুকূল(unsigned integer) হয়।

(iii). pc=&c; ভ্যারিয়েবল c এর এক্সেসকে পয়েন্টার pc তে এসাইন করে। যখন প্রিন্ট করা হয় তখন দেখবেন পয়েন্টার pc এর এক্সেস এবং c এর এক্সেস একই হবে। এছাড়া পয়েন্টার pc এর কন্টেন্টও একই(22) হবে।

(iv). c=11; ভ্যারিয়েবল c এ 11 এসাইন করে।

আমরা ভ্যারিয়েবল c এ নতুন ভ্যালু এসাইন করলাম যা পয়েন্টার pc কে প্রতিবিত করে।

(v). যেহেতু পয়েন্টার pc এবং c একই এক্সেসকে নির্দেশ করে, সুতরাং পয়েন্টার pc একই ভ্যালু 11 কেই রেফার করে। pc এর এক্সেস এবং কন্টেন্ট প্রিন্ট করলে আপনেও ভ্যালু 11 প্রদর্শিত হয়।

(vi). \*pc=2; পয়েন্টার pc বাবা নির্দেশিত মেমোরি লোকেশনের পূর্বের কন্টেন্ট পরিবর্তন করে 2 এসাইন করে।  
যেহেতু পয়েন্টার pc এবং ভ্যারিয়েবল c একই এক্সেসকে নির্দেশ করে সুতরাং c এর ভ্যালু পরিবর্তন হয়ে 2 হয়।

পয়েন্টার ব্যাবহারের সময় সাধারণ ভ্লকসমূহ:

ধরলে, পয়েন্টার pc কে c এর এক্সেস পয়েন্ট/নির্দেশ করতে চাচ্ছে।  
তাহলে

```

int c, *pc;
pc=c; /* ভ্লক! যেখানে pc হলো এক্সেস অর্থ এক্সেস নয়। */
*pc=&c; /* ভ্লক! যেখানে *pc হলো এক্সেসকে নির্দেশকৃত ভ্যালু অর্থ c হলো এক্সেস। */
pc=&c; /* Correct! যেহেতু pc এবং &c উভয়ই এক্সেস। */
*pc=c; /* সঠিক! যেহেতু *pc হলো এক্সেসকে নির্দেশকৃত ভ্যালু এবং c ও ভ্যালু। */
উভয় ক্ষেত্রেই পয়েন্টার pc, c এর এক্সেসকে নির্দেশ করে না।

```

সি প্রোগ্রামিং পয়েন্টার এবং আরো

**প্রশ্ন ১. Pointer array কি? চিত্রের মাধ্যমে দেখাও**  
**উত্তর:** Array কে যখন pointer হিসাবে declare করা হয় তখন তাকে pointer array বলে।

```

int x[5] = {1,2,3,4,5};
int * P = x;
x[0] x[1] x[2] x[3] x[4]
value→ 1 2 3 4 5
1000 1002 1004 1006 1008

```

এখানে P কে যদি Pointer হিসাবে declare করা হয় হলো:

P = & x[0]
P → x[0] 1 1000
2 1002
3 1004 ← Address
4 1006
5 1008

এখানে P, x[0] এর [0] location point করে ফলে P এর 1000 location টি point করে যার মধ্যে 1 value সংরক্ষিত আছে।  
Example:

```
#include <stdio.h>
int main () {
    int var = 20;
    int *ip;
    ip = &var;
    printf("Address of var variable: %x\n", &var );
    printf("Address stored in ip variable: %x\n", ip );
    printf("Value of *ip variable: %d\n", *ip );
    return 0;
}
```

**Output:**

Address of var variable: 4306dd94  
 Address stored in ip variable: 4306dd94  
 Value of \*ip variable: 20

**NULL pointer:** A pointer that is assigned NULL is called a null pointer.

The NULL pointer is a constant with a value of zero defined in several standard libraries.

**Example:**

```
#include <stdio.h>
int main () {
    int *ptr = NULL;
    printf("The value of ptr is : %x\n", ptr );
    return 0;
}
```

**Output:** The value of ptr is : 0

সি প্রোগ্রামিং পয়েন্টার এবং আরে: সি প্রোগ্রামিং এ array এবং pointer প্রয়োজন করে। কিন্তু এদের মধ্যে কৃতিত্বপূর্ণ পার্থক্য হলো পয়েন্টারের ভ্যালু হিসাবে বিভিন্ন এড্রেস থাকতে পারে পক্ষতরে array এর এড্রেস ফিক্সড(fixed) থাকে।

মিচের উদাহরণের সাহায্যে এদের মধ্যে পার্থক্য দেখানো হলো:

```
#include <stdio.h>
int main(){
    char charArr[4];
    int i;
    for(i = 0; i < 4; ++i){
        printf("Address of charArr[%d] = %u\n", i,
        &charArr[i]);
    }
    return 0;
}
আউটপুট:
Address of charArr[0] = 28ff44
Address of charArr[1] = 28ff45
Address of charArr[2] = 28ff46
Address of charArr[3] = 28ff47
নোটঃ আপনি array এর বিভিন্ন এড্রেস পেতে পারেন।
```

উপরের উদাহরণের ফলাফলে লক্ষ্য করলে দেখবেন যে, charArr আরে এর পরিপর দুটি এলিমেন্টের পার্থক্য সমান। অর্থাৎ এদের পার্থক্য হলো 1 বাইট। কিন্তু পয়েন্টার যেহেতু অন্য ভ্যারিয়েবলের লোকেশনকে নির্দেশ করে, সুতরাং ইহা যেকোনো এড্রেস জমা(store) করতে পারে।

সি প্রোগ্রামিং Array এবং Pointer এর মধ্যে সম্পর্ক নিচের ন্যায় একটি array বিবেচনা করুন:

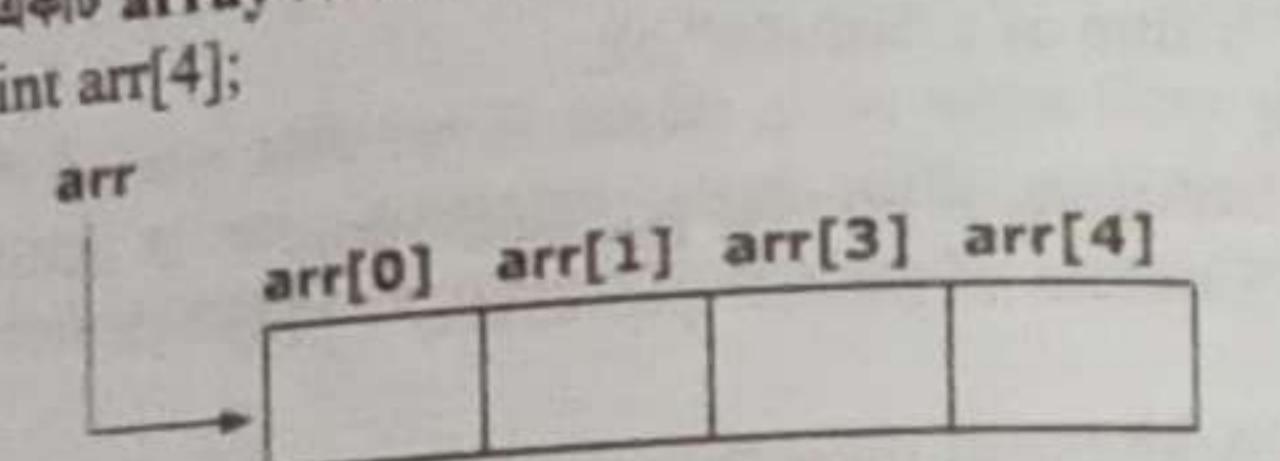


Figure: Array as Pointer

সি প্রোগ্রামিং এ array এর নাম সবসময় arr এর প্রথম এলিমেন্টের এড্রেসকে নির্দেশ করে।

উপরের প্রয়োজনীয় arr এবং &arr[0] উভয়ই প্রথম এলিমেন্টের এড্রেসকে নির্দেশ(point) করে।

```
&arr[0] সাধারণত arr এর সমান  

যেহেতু উভয়ের এড্রেস একই, সুতরাং arr এবং &arr[0] এর ভ্যালুও  

একই।  

arr[0] সাধারণত *arr(পয়েন্টারের এড্রেসের ভ্যালু) এর সমান।
```

একইভাবে,  
&arr[1] সাধারণত (arr + 1) এর সমান এবং arr[1] সাধারণত  
\*(arr + 1) এর সমান।  
&arr[2] সাধারণত (arr + 2) এর সমান এবং arr[2] সাধারণত  
\*(arr + 2) এর সমান।  
&arr[3] সাধারণত (arr + 1) এর সমান এবং arr[3] সাধারণত  
\*(arr + 3) এর সমান।

&arr[i] সাধারণত (arr + i) এর সমান এবং arr[i] সাধারণত  
\*(arr + i) এর সমান।

সি প্রোগ্রামিং এ array ডিক্লেয়ার(declare) করতে পারেন এবং  
পয়েন্টার ব্যবহার করে এই array এর ডেটা(data) পরিবর্তন করতে  
পারেন।

উদাহরণঃ Array এবং Pointer ব্যবহার করে ৫ টি পূর্ণ সংখ্যার  
যোগফল নির্ণয়ের প্রয়োগ।

```
#include <stdio.h>
int main(){
    int i, classes[5], sum = 0;
    printf("Enter 5 numbers:\n");
    for(i = 0; i < 5; ++i){ // (classes + i) সাধারণত  

        &classes[i] এর সমান  

        scanf("%d", &classes[i]); // *(classes + i) সাধারণত  

        classes[i] এর সমান  

        sum += *(classes + i);
    }
}
```

```
printf("Sum = %d", sum);
return 0;
}
```

আউটপুট:  
Enter 5 numbers: 5 4 10 5 7  
Sum = 31

❖ পয়েন্টারের রেফারেন্স কর্তৃক ফাংশন কল করা - C Call by

**Reference: Using pointers**

এই অধ্যায়ে পয়েন্টারকে ফাংশনের আর্গুমেন্ট(argument) হিসাবে  
অতিক্রম(pass) করানো শিখবেন। এছাড়া প্রয়াম্যে সঠিকভাবে পয়েন্টার  
ব্যবহার করা শিখবেন।

পয়েন্টারের রেফারেন্স কর্তৃক ফাংশন কল করা ফাংশন এর মধ্যে দিয়ে  
pointer কে যখন আর্গুমেন্ট হিসাবে অতিক্রম করানো হয় তখন ভ্যালুর  
পরিবর্তে মেমোরি লোকেশনের এড্রেস অতিক্রম হয়।

করলে সি প্রোগ্রামিং এ পয়েন্টার ভ্যালু জমা রাখার পরিবর্তে মেমোরির  
লোকেশনকে ভ্যালু হিসাবে জমা(store) রাখে।

উদাহরণঃ পয়েন্টার এবং ফাংশনের ব্যবহার রেফারেন্স(reference)  
হারা ফাংশন কল করে দুটি সংখ্যাকে বিনিময়(swap) করার জন্য সি  
প্রোগ্রাম।

```
/* পয়েন্টার এবং ফাংশন ব্যবহার করে দুটি সংখ্যা বিনিময়(swap) এর  

জন্য সি প্রোগ্রাম*/  

#include <stdio.h>
void swap(int *n1, int *n2);
int main(){
    int num1 = 5, num2 = 10; // num1 এবং num2 এর  

    এড্রেসকে swap() ফাংশনের মধ্যে দিয়ে অতিক্রম করানো হয়।
}
```

swap( &num1, &num2);

printf("Number1 = %d\n", num1);

printf("Number2 = %d", num2);

return 0;

}

void swap(int \* n1, int \* n2){ // পয়েন্টার n1 এবং n2  
যখনকামে num1 এবং num2 এর এড্রেসকে নির্দেশ(point) করে।

int temp;

temp = \*n1;

\*n1 = \*n2;

\*n2 = temp;

}

আউটপুট:

Number1 = 10

Number2 = 5

উপরের উদাহরণের ব্যাখ্যাঃ

(i). num1 এবং num2 ভ্যারিয়েবলের মেমোরি লোকেশনের এড্রেসকে  
swap() ফাংশনের মধ্যে দিয়ে অতিক্রম করানো হয়। পয়েন্টার \*n1  
এবং \*n2 এর এড্রেস-সমূহকে ভ্যালু হিসাবে এবং করে।

(ii). সুতরাং এখন পয়েন্টার n1 এবং n2 যখনকামে num1 এবং  
num2 এর এড্রেসকে নির্দেশ(point) করে।

(iii). পয়েন্টারের ভ্যালু যখন পরিবর্তন হয় তখন মেমোরি লোকেশনে  
নির্দেশিত(pointed) ভ্যালুও যথাক্রমে পরিবর্তন হয়।

(iv). সুতরাং পয়েন্টার \*n1 এবং \*n2 তে পরিবর্তন হটলে main()  
ফাংশনের num1 এবং num2 তেও পরিবর্তন হটে।

এই কৌশলকে সি প্রোগ্রাম এ রেফারেন্স কর্তৃক(Call by  
Reference) কল করা বলা হয়।

**প্রশ্ন 2. Pointer এর ব্যবহার লিখ।****উত্তর: Pointer এর ব্যবহার-**

- Function এর parameter হিসাবে।
- Function এর return value হিসাবে।
- Function pointer হিসাবে।

**প্রশ্ন 3. Pointer ব্যবহারের সুবিধা :** [NTRCA-2012, 2014]

- Pointers allows passing of arrays and strings to functions more efficiently.
- Pointer এর মাধ্যমে একটি function থেকে একাধিক value return পাওয়া সম্ভব।
- Pointer program এর জটিলতা হ্রাস করে।
- Pointers processing speed বৃদ্ধি করে।
- Pointer memory save করে।

**প্রশ্ন 4. Pointer এর অসুবিধা তালো লিখ।****উত্তর: Pointer এর অসুবিধা-**

- Pointer ব্যবহীক variable হতে দীর্ঘ গতিসম্পর্ক।
- Syntax জটিল।
- Pointer debug করা জটিল।
- ভাল ভাবে বোঝার জন্য সময় সাপেক্ষ।

**প্রশ্ন 5. START এবং Null pointer এর কাজ কি?**

উত্তর: START list এর প্রথম উপাদানকে Point করে। এবং  
NULL list সমষ্টিকে ঘোষণা করে।

**Self study****প্রশ্ন 1. List pointer variable কি?****প্রশ্ন 2. NULL pointer এবং NULL list এর মধ্যে পার্থক্য লিখ।**

ডাইনামিক মেমোরি এলোকেশন (সি)

Dynamic Memory Allocation (C)

**■ Dynamic Memory Allocation:**

Function	Syntax
malloc()	malloc (number*sizeof(int));

calloc()	calloc (number, sizeof(int));
realloc()	realloc (pointer_name, number * sizeof(int));
free()	free (pointer_name);

[malloc function execution এর সময় মেমোরি বরাদ্দ করার জন্য ব্যবহার করা হয়।]

ডাইনামিক মেমোরি এলোকেশন হলো কিছু লাইব্রেরী ফাংশন দেখন যেমন malloc(), calloc(), free() এবং realloc() ব্যবহার করে প্রোগ্রামে ডাইনামিকভাবে মেমোরি বরাদ্দ(dynamically allocate memory) করা।

সি প্রোগ্রাম এ, কম্পাইল টাইমে না ধারণা পর্যন্ত array এর সাইজ অজ্ঞান থাকে। অর্থাৎ এ সময় পর্যন্ত যতক্ষণ না কম্পাইলার কোডে কম্পিউটার বুঝে এমন কোডে এ (language) জুগাড় করে। সুতরাং মাঝে মাঝে array এর সাইজ (size) কম অথবা প্রয়োজনের তুলনায় বেশি হতে পারে। ডাইনামিক মেমোরি এলোকেশন (Dynamic memory allocation) প্রোগ্রাম run করার সময় হয় অধিক মেমোরি ধরে রাখে অথবা প্রয়োজন না হলে ছেড়ে দেয়।

সহজ অর্থে, ডাইনামিক মেমোরি এলোকেশন প্রোগ্রামের মেমোরি স্পেস manually পরিচালনা করার সম্ভবতি দেয়।

সি প্রোগ্রামে উভয়াধিকারে সূত্রে মেমোরি বরাদ্দের জন্য ডাইনামিক কোনো কোশল না ধারণেও হেডোর ফাইল "stdlib.h" এর অধীনে ডাইনামিকভাবে মেমোরি এলোকেশন এর জন্য ৪টি লাইব্রেরী ফাংশন রয়েছে।

#### ফাংশনের ব্যবহার:

malloc(): অন্যোদের ভিত্তিতে মেমোরি স্পেস বরাদ্দ করে এবং বরাদ্দ করা মেমোরির প্রথম বাইটের জন্য রিটার্ন করে।

calloc(): Array এর এলিমেন্টের জন্য মেমোরি স্পেস বরাদ্দ করে এবং এর প্রাথমিক ভ্যালু ০(শূন্য) থাকে। তারপরে মেমোরিতে রিটার্ন করে।

free(): পূর্বের বরাদ্দ করা মেমোরি স্পেস মুক্ত/ধালি করে।

realloc(): পূর্বের বরাদ্দ করা মেমোরি স্পেসের পরিবর্তন ঘটায়।

সি malloc() ফাংশন: malloc এর পূর্ণরূপ "memory allocation"।

malloc() ফাংশন মেমোরির একটি নির্দিষ্ট জায়গা (block) দখল করে এবং void টাইপের pointer রিটার্ন করে যার মাধ্যমে যেকোনো আকৃতির (from) পয়েন্টার নিক্ষেপ (cast) হতে পারে।

malloc() এর সিন্ট্যাক্স: ptr = (cast-type\*) malloc(byte-size)

এখানে ptr হলো নিক্ষেপ (cast) টাইপের। malloc() ফাংশন মেমোরির মধ্যে বাইট এককে নির্দিষ্ট সাইজের একটি রিটার্ন করে। মেমোরি স্পেস পর্যাপ্ত না হলে NULL রিটার্ন (return) করে।

ptr = (int\*) malloc(50 \* sizeof(int));

এই স্টেটমেন্টটি int এর সাইজ (2 অথবা 4 বাইট) এর উপর ভিত্তি করে যথাক্রমে 100 অথবা 200 বাইট মেমোরি বরাদ্দ করতে পারে এবং পয়েন্টার মেমোরির প্রথম বাইটের এক্সেস রেফার (point) করে।

সি calloc() ফাংশন: calloc এর পূর্ণরূপ "contiguous allocation"।  
malloc() এবং calloc() এর মধ্যে তথ্যাত্মক পার্থক্য হলো malloc() ফাংশন মেমোরির সিঙ্গেল ব্রক (single block of memory) বরাদ্দ করে পক্ষতরে calloc() ফাংশন মেমোরির মাল্টিপ্ল ব্রক (multiple blocks of memory) বরাদ্দ করে এবং প্রত্যেকের সাইজ সেট হয় ০(শূন্য)।  
calloc() এর সিন্ট্যাক্স: ptr = (cast-type\*)calloc(n, element-size);

এই স্টেটমেন্টটি n সংখ্যক এলিমেন্ট বিশিষ্ট array জন্য সন্তুষ্টিত (contiguous) মেমোরি স্পেস বরাদ্দ করবে। উদাহরণস্বরূপঃ  
ptr = (float\*) calloc(50, sizeof(double));

এই স্টেটমেন্টটি ৫০টি এলিমেন্ট বিশিষ্ট array জন্য double(8 bytes) সাইজের সন্তুষ্টিত (contiguous) মেমোরি স্পেস বরাদ্দ করবে।

সি free() ফাংশন: calloc() অথবা malloc() ফাংশন কর্তৃক ডাইনামিকভাবে (Dynamically) বরাদ্দ হওয়া মেমোরি নিজে নিজে মুক্ত হতে পারে না। তাই আটককৃত (allocated) মেমোরি বালাস (release) করার জন্য অবশ্যই free() ফাংশন ব্যবহার করতে হবে।

free() সিন্ট্যাক্স: free(ptr);

এই স্টেটমেন্টটি পয়েন্টার (ptr) কর্তৃক মেমোরিতে বরাদ্দ হওয়া স্পেসকে খালি করে।

উদাহরণ সি malloc() এবং free() ফাংশনের ব্যবহার: ইউজার কর্তৃক ইনপুট নিয়ে n সংখ্যক এলিমেন্টের যোগফল নির্ণয়ের জন্য সি প্রোগ্রাম। এই প্রোগ্রামটি সম্পাদনের জন্য calloc() ফাংশন ব্যবহার করে ডাইনামিকভাবে মেমোরি বরাদ্দ (allocate) করতে হবে।

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int num, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &num);
    ptr = (int*) malloc(num * sizeof(int));
    //malloc ব্যবহার করে মেমোরি বরাদ্দ করা
    if(ptr == NULL){
        printf("Error! memory not allocated.");
        exit(0);
    }
    printf("Enter elements of array: ");
    for(i = 0; i < num; ++i){
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
    printf("Sum = %d", sum);
    free(ptr);
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int num, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &num);
    ptr = (int*) malloc(num * sizeof(int));
    //malloc ব্যবহার করে মেমোরি বরাদ্দ করা
    if(ptr == NULL){
        printf("Error! memory not allocated.");
        exit(0);
    }
    printf("Enter elements of array: ");
    for(i = 0; i < num; ++i){
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
    printf("Sum = %d", sum);
    free(ptr);
    return 0;
}
```

#### Output:

Enter number of elements: 5  
Enter elements of array: 1 2 3 4 5  
Sum = 15

উদাহরণ সি calloc() এবং free() ফাংশনের ব্যবহার: ইউজার কর্তৃক ইনপুট নিয়ে n সংখ্যক এলিমেন্টের যোগফল নির্ণয়ের জন্য সি প্রোগ্রাম লিখুন। এই প্রোগ্রামটি সম্পাদনের জন্য calloc() ফাংশন ব্যবহার করে ডাইনামিকভাবে মেমোরি বরাদ্দ (allocate) করতে হবে।

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int num, i, *ptr, sum = 0;
    printf("Enter number of elements: ");
    scanf("%d", &num);
    ptr = (int*) calloc(num, sizeof(int));
    //calloc ব্যবহার করে মেমোরি বরাদ্দ করা
    if(ptr == NULL){
        printf("Error! memory not allocated.");
        exit(0);
    }
    printf("Enter elements of array: ");
    for(i = 0; i < num; ++i){
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
    printf("Sum = %d", sum);
    free(ptr);
    return 0;
}
```

#### Output:

Enter number of elements: 10  
Enter elements of array: 1 2 3 4 5 6 7 8 9 10  
Sum = 55

সি realloc() ফাংশন: ইতিমধ্যে বরাদ্দকৃত মেমোরির সাইজ যদি প্রয়োজনের তুলনায় কম বা বেশি হয় তাহলে realloc() ফাংশন ব্যবহার করে আপনি পূর্ববর্তী মেমোরির সাইজ পরিবর্তন করতে পারেন।

realloc() এর সিন্ট্যাক্স: ptr = realloc(ptr, newsize);  
এখনে ptr হলো নতুন সাইজের বরাদ্দ হওয়া মেমোরি।

#### উদাহরণ সি realloc() ফাংশনের ব্যবহার:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int *ptr, i, n1, n2;
    printf("Enter size of array: ");
    scanf("%d", &n1);
    ptr = (int*) malloc(n1 * sizeof(int));
    printf("Address of previously allocated memory: ");
    for(i = 0; i < n1; ++i)
        printf("%u\t", ptr + i);
}
```

```
printf("%u\t", ptr + i);
printf("\nEnter new size of array: ");
scanf("%d", &n2);
ptr = realloc(ptr, n2);
for(i = 0; i < n2; ++i)
    printf("%u\t", ptr + i);
return 0;
}
```

#### Output:

Enter size of array: 5  
Address of previously allocated memory: 7015792 7015796 7015800 7015804 7015808  
Enter new size of array: 6  
7015792 7015796 7015800 7015804 7015808 7015812

1. Why does every one want to dynamically allocate memory? What is the benefit of doing so?  
[প্রযোজনী সম্পাদন ও প্রোগ্রাম কর্মসূচী কর্মসূচী বৃত্তো-২০১৮ ইলেক্ট্রোটেকনিক্স]

উত্তরঃ ডাইনামিক মেমরি আলোকেশন প্রযোজন হয় যখন worst case এ মেমরি requirement জানা না থাকে। statically মেমরি allocate করা অস্বীকৃত হলে dynamically মেমরি allocate করা হয়। unused memory জন্য অতিরিক্ত খরচ নেই। ডাইনামিক মেমরি এলোকেশন (Dynamic memory allocation) প্রযোজন run করার সময় হয় অধিক মেমরি ধরে রাখে অথবা প্রযোজন না হলে ছেড়ে দেয়।

ইহার মাধ্যমে মেমরিকে reallocation করা সহজ। বা অন্য কথায়, একই আরের জন্য একটি new fragment বরাদ্দ করা সুবিধাজনক, এবং additional memory allocate করা যায় বা unnecessary memory release করা যায়।

2. How does dynamic memory allocation help in managing data?

উত্তরঃ মেমরিকে reallocation করা সুবিধাজনক। বা অন্য কথায়, একই আরের জন্য একটি new fragment বরাদ্দ করা সুবিধাজনক, এবং additional memory allocate করা যায় বা unnecessary memory release করা যায়। তাই dynamic memory allocation মেমরি ম্যানেজমেন্ট/ডাটা ম্যানেজমেন্ট এ হেল্প করে।

#### Structure

Structure হল user defined data type। নিজস্ব data type তৈরির জন্য structure declare করা হয়। [NTRCA-2010,17]

#### Syntax:

```
struct structureName{
    //member definitions
};
```

#### Example:

```
struct Courses{
    char WebSite[50];
    char Subject[50];
    int Price;
};
```

### Structure Programming (C Program)

```
■ Structure ব্যবহার করে দুইটি দূরত্ব যোগ করার প্রোগ্রাম :
struct Distance{
    int feet;
} d1, d2, sumOfDistances;
int main(){
    printf("Enter information for 1st
           distance in feet\n");
    scanf("%d", &d1.feet);
    printf("\nEnter information for 2nd
           distance in feet\n");
    scanf("%d", &d2.feet);
    sumOfDistances.feet = d1.feet+d2.feet;
    printf("\nSum of distances = %d"
           ,sumOfDistances.feet);
    return 0;
}
```

#### Output:

Enter information for 1st distance in feet: 15  
 Enter information for 2nd distance in feet:20  
 Sum of distances = 35

#### ■ Differences Between Structure and Union [NTRCA-2011, BOF-SAE-2021]

Union	Structure
Uses keyword "union"	Uses Keyword "struct"
Union এর Memory size সবথেকে বড় element এর Memory size এর সমান।	Structure এর Memory size সব element এর Memory size এর যোগফলের সমান।
Union এর element গুলো নিজেদের মধ্যে Memory Share করে।	Structure এর element গুলো নিজেদের মধ্যে Memory Share করে না।
একসাথে তথ্যাত্মক একটি element access করা যায়।	একসাথে একাধিক element access করা যায়।
Union example	struct example
{ Data Types and members }obj;	{ Data Types and members }obj;

#### ■ C Programming Language এ Structure & Array

এর মধ্যে পার্থক্য লিখুন। [সার্ভিজ মানসুন- ২০২১-সহকারী প্রোগ্রামার (মন-  
ক্যাডের), BOF-AE-2021]

উত্তর: Structure & Array এর মধ্যে পার্থক্য:

Structure	Array
১. বিভিন্ন ডেটা টাইপের ভেরিয়েবলের সংগ্রহকে Structure বলে।	১. একই ডেটা টাইপের ভেরিয়েবলের সংগ্রহকে Array বলে।
২. Structure এর উপাদানগুলি একটি contiguous memory তে সংরক্ষণ করা যায় না।	২. Array এর উপাদানগুলি একটি contiguous memory তে সংরক্ষণ করা হয়।

৩. Structure এর উপাদানগুলি তাদের নাম দ্বারা আস্রেস করা হয়।	৩. অ্যারের উপাদানগুলি index নম্বর দ্বারা আস্রেস করা হয়।
৪. Structure এর প্রতিটি উপাদান বিভিন্ন ডেটা টাইপের হতে পারে।	৪. অ্যারের প্রতিটি উপাদান একই আকারের।
e. Syntax: struct struct name{ type element; type element; }; variable1, variable2, . . .;	e. Syntax: type array_name[size];

◆ Write a program to read the coordinates of the end points of line and to find its length. Use a structured variable name line to store the relevant information about its end points. [স্ট্রাকচার ব্যবহার করে,  
একটি লাইনের দৈর্ঘ্য বের করার জন্য সি প্রোগ্রাম লিখ, যেখানে একটা  
লাইনের দুটি ইতেক ইনপুট নিতে হবে ] [Multiple Ministry -  
assistant programmer- 2017]

```
#include<stdio.h>
#include<string.h>
typedef struct calculate{
    float crd;
    float length;
}c;

int main(){
    c x1, x2, y1, y2, temp;
    printf("Enter x-crd of the first point : ");
    scanf("%f", &x1.crd);
    printf("Enter y-crd of the first point : ");
    scanf("%f", &y1.crd);
    printf("Enter x-crd of the second point : ");
    scanf("%f", &x2.crd);
    printf("Enter y-crd of the second point : ");
    scanf("%f", &y2.crd);
    temp.length=sqrt((x2.crd-x1.crd)*(x2.crd-  
x1.crd)+(y2.crd-y1.crd)*(y2.crd-y1.crd));
    printf("\nLength : %f", temp.length);
    return 0;
}
```

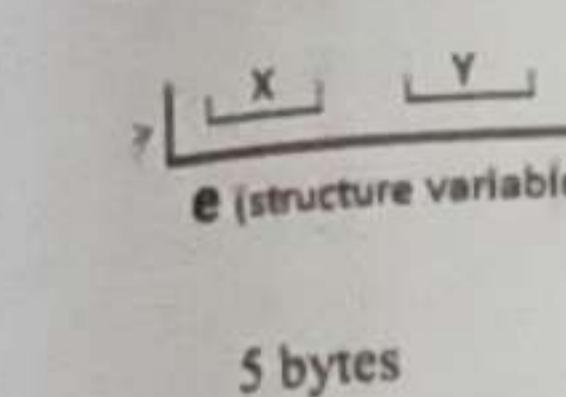
#### ■ Union ■

Unions হল user defined data type. Unions এর structures একই, পার্থক্য তখন storage system এ. Union  
এর Memory size সবথেকে বড় element এর Memory  
size এর সমান। Union এর element গুলো নিজেদের মধ্যে  
Memory Share করে। [NTRCA-2014,17]

### Structure Programming (C Program)

#### Structure

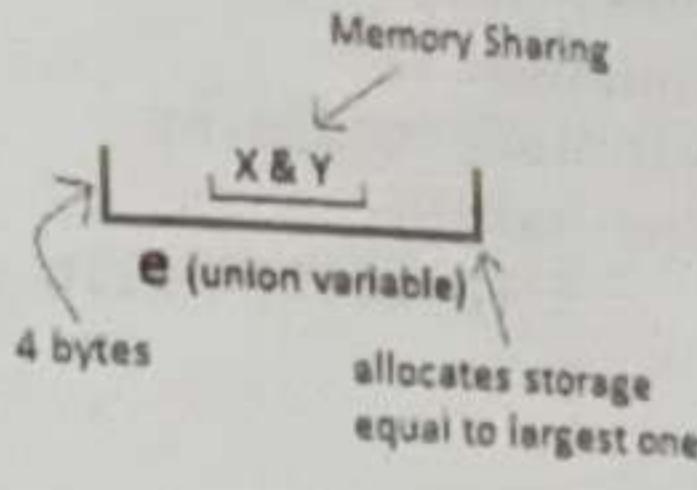
```
struct Emp
{
    char X; // size 1 byte
    float Y; // size 4 byte
};
```



5 bytes

#### Unions

```
union Emp
{
    char X;
    float Y;
};
```



4 bytes

#### FILE I/O

C programming এ file এর I/O function, secondary storage device এ ডাটা access এর জন্য ব্যবহৃত হয়।

C File Operations প্রধানত 5 ধরনের Operation সম্পর্কে:

- > Create : নতুন ফাইল তৈরি করা।
- > Open : পুরাতন ফাইল ওপেন করা।
- > Read : ফাইল থেকে ডাটা পড়া।
- > Write : ফাইল এ ডাটা লিখা।
- > Close: ফাইল বন্ধ করা।

#### Steps for Processing a File :

- > file pointer variable Declare করতে হবে।
- > ফাইল ওপেন করা জন্য fopen() function ব্যবহার করা  
হয়।
- > প্রযোজনীয় অপারেশন সম্পর্ক করা।
- > fclose() function ব্যবহার করে ফাইল বন্ধ করা।

#### ■ ফাইল কে পরিচালনা করার জন্য শাইট্রেরি function গুলো হলো

Function	Syntax:	Uses/Purpose
fopen()	FILE *fopen( const char * filePath, const char * mode);	ফাইল ওপেন করার জন্য
fclose()	fclose(fp);	ফাইল বন্ধ করার জন্য
getc()	getc();	ফাইল হতে একটি করে ক্যারেক্টোর লিখার জন্য
putc()	int fputc( int c, FILE * stream );	ফাইল এ একটি করে ক্যারেক্টোর লিখার জন্য
getw()	int getw( FILE * stream );	integer লিখ করার জন্য
putw()	int putw( int c, FILE * stream );	Integer লিখার জন্য

fprintf()	int fprintf(FILE *stream, const char *format, ...)	ফাইল এ formatted output Prints করার জন্য
fscanf()	int fscanf(FILE *stream, const char *format, ...)	ফাইল এ formatted input এর জন্য
fgets()	char *fgets(char *str, int n, FILE *stream)	ফাইল থেকে স্ট্রিং রিড করার জন্য
fputs()	int fputs(const char *str, FILE *stream)	ফাইল থেকে স্ট্রিং write করার জন্য
feof()	int feof(FILE *stream)	end of file পর্যন্ত রিড করার জন্য

#### ■ ফাইল এর বিভিন্ন অকার mode:

Mode	Description
R	অধৃত করার জন্য।
r+	ফাইল এ read and write operation করার জন্য [open for reading and writing from beginning]
W	লিখার জন্য।
w+	ফাইল এ রিডিং অ্যাড writing করার জন্য [open for reading and writing, overwriting a file]
A	ফাইল এ শেষে লিখার জন্য।
a+	Open for reading and appending and create the file if it does not exist. The reading will start from the beginning but writing can only be appended.

#### The main difference between w+ and r+:

-w+ truncate the file to zero length if it exists or create a new file if it doesn't.

-While r+ neither deletes the content nor create a new file if it doesn't exist.

#### Example:

```
#include<stdio.h>
int main(){
    FILE *fp = fopen("fileName.txt", "r");
    int ch = getc(fp);
    while (ch != EOF) {
        putchar(ch);
        ch = getc(fp);
    }
    if (feof(fp))
```

```

        printf("\n Reached the end of
file.");
else
    printf("\n Something gone wrong.");
fclose(fp);
return 0;
}

```

[এখনে program টি c ভাষার স্নেহ তাই একটু বিনিয়ন মনে হতে পারে তবে কাইল সম্পর্কিত প্রয়োগ Example অধীনে সহজ তাবে দেখা আছে]

#### প্রশ্ন ১. EOF কী?

আমরা যখন কোনো উৎস বা সোর্স থেকে তাঁটা ইনপুট গ্রহণ করে, প্রয়োগিকভাবে সেই উৎস বা সোর্সকে বলা হয় stream বা file. যখন কোনো তাঁটা সোর্স থেকে আর কোনো তাঁটা ইনপুট নেওয়া যাবেন (এটা অনেক কারণে হতে পারে, আর কোনো তাঁটা নই অথবা, ডিভাইস ডিসকনেনেক্ট হওয়া অথবা লেটার্যাক কানেকশন বিছিন্ন হওয়া) তখন সেই অবস্থা বোকানের জন্য একটি সিগনাল তৈরী করা হয়। সেটা বেই EOF বা End of File বলে।

**Android** কোনের এখন অনেক IDE পাওয়া যাব যাব সাবাদে এই সকল প্রয়োগ RUN করে দেখতে পারেন, এতে করে স্বতে আরও সহজ হবে। [কিউ সিপি Compiler(IDE) এর নাম: CppDroid - C/C++ IDE, c4droid (c/c++ compiler), Online Compiler.

#### Programming Part General

Character	ASCII VALUE
A-Z	65-90
a-z	97-122
0-9	48-57

#### প্রয়োগ ১. ASCII value পিছে করার সি প্রয়োগ লেখ।

```

#include<stdio.h>
int main(){
    int i;
    for(i=0;i<=255;i++)
        printf("ASCII value of character
%c: %d\n",i,i);
    return 0;
}

```

#### প্রয়োগ ২. যেকোন একটি সংখ্যা এর ASCII value পিছে করার সি

```

#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
int main(){
    char c;
    printf("Enter any character: ");
    scanf("%c",&c);
    printf("ASCII value of given
character: %d",c);
}

```

```

    return 0;
}

```

#### Sample output:

Enter any character: a  
ASCII value of given character: 97

#### প্রয়োগ ৩. Semicolon ব্যবহার না করে Hello World প্রিণ্ট করার সি প্রয়োগ লেখ।

#### Solution: 1

```

#include<stdio.h>
int main(){
    if(strcmp("Hello world")){
    }
}

```

#### Solution: 2

```

#include<stdio.h>
int main(){
    while(!strcmp("Hello world")){
    }
}

```

#### Solution: 3

```

#include<stdio.h>
int main(){
    switch(strcmp("Hello world")){
    }
}

```

#### প্রয়োগ ৪. Farenheit থেকে Celcius Convert করার জন্য c

program লিখ।[PGCB-2018, Bangabandhu Sheikh Mujibur Rahman Aviation and Aerospace University - AP&ASE - 2019]

```

#include<stdio.h>
int main(){
    float c,f;
    printf("Enter farenheit : ");
    scanf("%f",&f);
    c = ((f - 32) * 5)/9;
    printf("celcius: %f",c);
    return 0;
}

```

#### Sample Output:

Enter farenheit: 98  
celcius: 36.666668

#### প্রয়োগ ৫. Feet থেকে Meter Convert করার জন্য c ভাষায়

program লিখ।[HW: Hints 1 ft = 3.28 m]

#### প্রয়োগ ৬. একটি সংখ্যা এর square root( $\sqrt{ }$ ) বের করার সি প্রয়োগ

লেখ।

```

#include<stdio.h>
#include<math.h>
int main(){
    int num;

```

```

    scanf("%d",&num);
    printf("Square root: %f",sqrt(num));
    return 0;
}

```

#### Sample: 5

Square root: 2.236068

#### প্রয়োগ ৭. বৃত্তের ক্ষেত্রফল নির্ণয়ের করার সি প্রয়োগ লেখ।

```

#include <stdio.h>
#define PI 3.141
int main(){
    float r, a;
    printf("Radius: ");
    scanf("%f", &r);
    a = PI * r * r;
    printf("%f\n", a);
    return 0;
}

```

#### Sample Output:

Radius: 3

28.268999

#### প্রয়োগ ৮. যেকোন একটি সংখ্যা থেকে শেষ দুটি সংখ্যা আলাদা করার সি

প্রয়োগ লেখ।

```

#include<stdio.h>
int main(){
    int yyyy,yy;
    printf("Enter a digits: ");
    scanf("%d",&yyyy);
    yy = yyyy % 100;
    printf("Last two digits is: %02d",yy);
    return 0;
}

```

#### Sample Output:

Enter a year in for digits: 1985  
Last two digits of year is: 85

#### প্রয়োগ ৯. সমকেলী ত্রিভুজের ক্ষেত্রফল নির্ণয়ের প্রয়োগ লিখ। (Δ)

```

#include<stdio.h>
int main(){
    float h,w;
    float area;
    printf("Enter height and width
triangle : ");
    scanf("%f%f",&h,&w);
    area = 0.5 * h * w;
    printf("Area of triangle
is:%.3f",area);
    return 0;
}

```

#### Sample output:

Enter height and width triangle: 10 5  
Area of triangle is: 25.000

#### প্রয়োগ ১০. ত্রিভুজের ক্ষেত্রফল নির্ণয়ের প্রয়োগ লিখ।(Δ)

[Bangladesh China Power Company Limited -

Junior Instructor (Computer) -2019]

```

#include <stdio.h>
#include <math.h>
void main(){
    int s, a, b, c, area;
    printf("Enter the values of a, b and c
\n");
    scanf("%d %d %d", &a, &b, &c);
    s = (a + b + c) / 2;
    area = sqrt(s * (s - a) * (s - b) * (s -
c));
    printf("Area of a triangle = %d \n",
area);
    return 0;
}

```

#### প্রয়োগ ১১. আরডকেজ এর ক্ষেত্রফল নির্ণয়ের প্রয়োগ লিখ।

[HW] Formula :Area = length \* width

#### প্রয়োগ ১২. ঘনক এর ক্ষেত্রফল নির্ণয়ের প্রয়োগ লিখ।

Formula of volume of cube:Volume =  $a^3$

#### প্রয়োগ ১৩. কনক এর আয়তন ও তলাটের ক্ষেত্রফল নির্ণয়ের প্রয়োগ

লিখ।  
Surface\_area = Pie \* r \* (r +  $\sqrt{r^2 + h^2}$ )  
Volume =  $1/3 * Pie * r^2 * h$

#### প্রয়োগ ১৪. Temporary variable ছাড়াই দুইটি সংখ্যা

বিনিয়োগ(Swapping) করার সি প্রয়োগ লেখ। [BUET IICT-2017, BREB - Assistant Junior Engineer (IT) -2019, BREB - ASE - 2019]

```

#include<stdio.h>
int main(){
    int a=5,b=10;
    //প্রতিঃ ০১:
    a=b+a;
    b=a-b;
    a=a-b;
    printf("a= %d b= %d",a,b);
}

```

#### প্রতিঃ ০২:

```

a=5;b=10;
a=a+b-(b-a);
printf("\na= %d b= %d",a,b);

```

#### প্রতিঃ ০৩:

```

a=5;b=10;
a=a^b;
b=a^b;

```

```
a=b;a;
printf("\na= %d b= %d",a,b);

প্রক্রিয়া:
a=5;b=10;
a=b~~a-1;
b=a~~b+1;
a=a~~b+1;
printf("\na= %d b= %d",a,b);

গ্রহণ ১৫. সূইটি সংখ্যা বিনিময়(Swapping) করার সি প্রোগ্রাম লিখো:
#include<stdio.h>
int main(){
    int a,b,c;
    scanf("%d %d",&a,&b);
    c = a;
    a = b;
    b = c;
    printf("%d %d",a,b);
    return 0;
}
```

**Sample output:** 4 6  
6 4

গ্রহণ ১৬. সূইটি সংখ্যা বিনিময়(Swapping) করার সি প্রোগ্রাম লেখ using pointer:
#include <stdio.h>
int main(){
 int x, y, \*a, \*b, temp;
 printf("Enter the value of x and y\n");
 scanf("%d%d", &x, &y);
 a = &x;
 b = &y;
 temp = \*b;
 \*b = \*a;
 \*a = temp;
 printf("After Swapping\nx = %d\ny = %d\n", x, y);
 return 0;
}

গ্রহণ ১৮. একটি ৯ ধার বিশিষ্ট বহুভুজের অফিচিয়াল ধার সমান। উক্ত বহুভুজের অভ্যন্তরীন কোন ভিত্তিতে একাশের সি প্রোগ্রাম লিখুন। [NPCBL - Junior Technical Engineer -2019]

```
#include<stdio.h>
int main(){
    int n, angle;
    printf("Enter the polygon side:");
    scanf("%d", &n);
    angle = ((n-2)*180)/n;
```

```
        printf("The angle is: %d",
angle);
return 0;
}

গ্রহণ ১৯. একটি সিসেল integer value ইনপুট নিয়ে উকাকে করার একাশ করার সি প্রোগ্রাম লেখ।
```

```
#include<stdio.h>
main(){
    int n,rem;
    printf("Enter number ");
    scanf("%d",&n);
    switch(n){
        case 0 : printf("Zero ");break;
        case 1 : printf("One ");break;
        case 2 : printf("Two ");break;
        case 3 : printf("Three ");break;
        case 4 : printf("Four ");break;
        case 5 : printf("Five ");break;
        case 6 : printf("Six ");break;
        case 7 : printf("Seven ");break;
        case 8 : printf("Eight ");break;
        case 9 : printf("Nine ");break;
    }
    return 0;
}
```

**Sample output:** Enter number 5  
Five

গ্রহণ ২০. সূইটি সংখ্যা যোগের সি প্রোগ্রাম লেখ যোগ অপারেটর (+) ব্যবহার না করে।  
**Algorithm:**

In  $c \sim i$  is I's complement operator. This is equivalent to:  
 $\sim a = -b + 1$  So,  $a \sim b - 1$   
 $= a(\sim b + 1) + 1 = a + b - 1 + 1 = a + b$

```
#include<stdio.h>
int main(){
    int a,b;
    int sum;
    printf("Enter any two integers: ");
    scanf("%d%d",&a,&b);
    sum = a - ~b - 1;
    printf("Sum of two integers: %d",sum);
    return 0;
}
```

**Sample output:**  
Enter any two integers: 5 10  
Sum of two integers: 15

গ্রহণ ২১. সূইটি সংখ্যা বিয়োগের সি প্রোগ্রাম লেখ বিয়োগ অপারেটর (-) ব্যবহার না করে।

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

### Conditional Programming

Program 1. Write a program check odd and even number using conditional operator.[ একটি নামুর জোড় এবং বিজোড় চেক করার জন্য প্রয়োগ লিখো] [Bangladesh China Power Company Limited - Junior Instructor (Computer) - 2019]

```
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    n%2 == 0 ? printf("Even Number") :
    printf("Odd Number");
    return 0;
}
```

Program 3. Convert Km to Meter, Centimeter or Millimeter.

```
#include <stdio.h>
int main() {
    char v;
    float km,ans=0;
    printf("enter Km :");
    scanf("%f",&km);
    printf("KM will be changed in (M,s,m) : ");
    scanf("%s",&v);
    if(v=='M')
        ans=km*1000;
    if(v=='S')
        ans=km*100000;
    if(v=='m')
        ans=km*1000000;
    printf("Answer : %.2f",ans);
}
```

**Sample Output:** 5

KM will be changed in (M,s,m) : m  
Answer : 5000000.00

Program 4. C program to find max of two numbers.

```
#include<stdio.h>
int main(){
    int a,b,c,big;
    printf("\nEnter 3 numbers:");
    scanf("%d %d %d",&a,&b,&c);
    big=(a>b&&a>c?a:b>c?b:c);
    printf("\nThe biggest number is:
%d",big);
    return 0;
}
```

**Sample Output:**

Enter 3 numbers: 9 87 63  
The biggest number is: 87

### Structure Programming (C Program)

**Sample Output:**

Enter the value of a and b : 45 63  
63

**Program 5. C program to check a given number is positive or negative.**

```
#include<stdio.h>
int main(){
    int num;
    printf("Enter a Number ");
    scanf("%d",&num);
    if(num == 0)
        printf("Number is zero.");
    else if(num > 0)
        printf("Number is positive.");
    else
        printf("Number is negative.");
    return 0;
}
```

**Sample Output:**

Enter a Number 45  
Number is positive.

**Program 6. C program to check whether triangle is valid or not if sides are given.**

```
#include <stdio.h>
int main(){
    int side1, side2, side3;
    printf("Enter three sides of triangle: \n");
    scanf("%d%d%d", &side1, &side2, &side3);
    if((side1 + side2 > side3) && (side1 + side3 > side2) && (side2 + side3 > side1)) {
        printf("Triangle is valid.");
    }
    else {
        printf("Triangle is not valid.");
    }
    return 0;
}
```

**Program 7. Write a program to accept a number from user and print if it is multiple of 7 in C language. [ একটি সংখ্যা ইউজার থেকে ইনপুট নিয়ে চেক করন . নথার টি ৭ এর অনিয়ত কিনা? ]**

```
#include<stdio.h>
#include<conio.h>
int main(){
    int n;
    printf("Enter number ");
    scanf("%d",&n);
}
```

```
if(n%7==0)
    printf("Number is multiple of 7");
else
    printf("Number is not multiple of
7");
return 0;
}
```

**Sample Output:**

Enter number 45  
Number is not multiple of 7

**Program 8. Program to Check Leap Year. [NTRCA-2013]**

```
#include <stdio.h>
int main() {
    int year;
    scanf("%d",&year);
    if (((year % 4 == 0) && (year % 100!=
0)) || (year%400 == 0))
        printf("%d is a leap year", year);
    else
        printf("%d is not a leap year",
year);
    return 0;
}
```

**Program 9. C program to calculate roots of a quadratic equation. [১৬ অক্ষয় নিবন্ধন (সিলিট) পরীক্ষা-২০১৯]**

```
#include <stdio.h>
#include <math.h>
int main() {
    double a,b,c,root1,root2;
    printf(" Please enter a: \n");
    scanf("%lf",&a);
    printf(" Please enter b: \n");
    scanf("%lf",&b);
    printf(" Please enter c: \n");
    scanf("%lf",&c);
    root1 = (-b + sqrt(b*b-4*a*c)) / (2*a);
    root2 = (-b - sqrt(b*b-4*a*c)) / (2*a);
    printf("\n First root is: %lf ",root1);
    printf("\n Second root is: %lf ",root2);
    printf("\n ");
    return 0;
}
```

**Looping Program ↗**
**Program 1. C program to print numbers from 1 to N.**

```
#include<stdio.h>
int main(){
```

### Structure Programming (C Program)

```
int i,n;
scanf("%d",&n);
for(i=1;i<=n;i++){
    printf("%d ",i);
}
return 0;
}
```

**Program 2. Write a C program to find the factor of n value. Where, Input: 10, Output: 7 (5,2) factor. [Dhaka Transport Co-ordination Authority - AP -2019]**

```
#include<stdio.h>
int main(){
    int n, i, sum=0;
    printf("Enter a positive integer for
find factor summation: ");
    scanf("%d", &n);
    int arr[n], p=0;
    printf("Factors are:");
    for(i=2; i<n; i++){
        if(n%i==0){
            printf("%d ", i);
            sum+=i;
        }
    }
    printf("\nFactor summation is:
%d\n", sum);
    return 0;
}
```

**Program 3. Summation of this series 1 + 2 + 3 + 4**

```
+ ... + n.
int main(){
    int i,n,sum=0;
    scanf("input: %d",&n);
    for(i=1;i<=n;i++){
        sum+=i;
    }
    printf("Sum: %d",sum);
    return 0;
}
```

**Sample output:**

input: 10  
Sum: 55

**Program 4. Using C Program to find the sum of the following series. 1+1/2+1/3+1/4+1/5+.....+1/N. [Univer Bank - SO (IT) -2019, ১৬ অক্ষয় নিবন্ধন (সিলিট) পরীক্ষা-২০১৯]**

```
#include<stdio.h>
#include<math.h>
int main(){
    int n;
    float i, sum=0.0;
    printf("Enter Nth position value
of the series: ");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
        sum+=(1/i);
    printf("Summation is : %f",
sum);
    return 0;
}
```

**Program 5. Write a C program to find max price from 20 items. [BD Bank - AP -2019, BD Computer Council 4 tier Data Center - AP -2019]**

```
#include<stdio.h>
int main(){
    int arr[20], i, max=0;
    printf("Enter 20 items price: ");
    for(i=0; i<20; i++)
        scanf("%d", &arr[i]);
    max=arr[0];
    for(i=1; i<20; i++) {
        if(arr[i]>max)
            max=arr[i];
    }
    printf("Maximum price is: %d", max);
    return 0;
}
```

**Program 6. How many bit have to change to convert int A to int B. sample A=10, B=20 output: 4 |একটি নথার A থেকে B তে পরিবর্তন করলে কতভাবে বিট পরিবর্তন হবে . আর কত প্রয়োগ লিখ? উদাহরণ: A=10( 01010) পরিবর্তন হবে যদি B=20(10100) হয়, আহলে 8 টা বিট পরিবর্তন হবে, যেই বিট সংখ্যা কোরে করতে হবে | [Basic Bank Ltd. - AM (ICT) -2019, Combined 3 banks, Combined (BHBFC & BKB) Assis. Programmer 2018, DBBL - Probationary Officer -2017]**

```
#include<stdio.h>
#include<math.h>
int main(){
    int a, b, c=0, p=0;
    printf("Enter 1st number : ");
    scanf("%d", &a);
    printf("Enter 2nd number : ");
    scanf("%d", &b);
    c=a^b;
    while(c) {

```



## Structure Programming (C Program)

```

        sum=sum+r;
    }
    printf("Sum of digits of number:
%d",sum);
    return 0;
}

```

**Sample output:**

Enter a number: 27421  
Sum of digits of number: 16

$$10|27421 \quad |2742 \quad 2742 \% 10 = 1 \quad 2742 / 10 = 274$$

$$\begin{array}{r} 2740 \\ \hline 1 \end{array}$$

$$10|2742 \quad |274 \quad 274 \% 10 = 2 \quad 274 / 10 = 27$$

$$\begin{array}{r} 2740 \\ \hline 2 \end{array}$$

$$10|274 \quad |27 \quad 27 \% 10 = 4 \quad 27 / 10 = 2$$

$$\begin{array}{r} 270 \\ \hline 4 \end{array}$$

$$10|27 \quad |2 \quad 2 \% 10 = 7 \quad 2 / 10 = 2$$

$$\begin{array}{r} 20 \\ \hline 7 \end{array}$$

$$10|2 \quad |0 \quad 2 \% 10 = 2 \quad 2 / 10 = 0$$

$$\begin{array}{r} 0 \\ \hline 2 \end{array}$$

**Program 16. How to calculate power( $x^y$ ) of a number in c.**
**Program:**

```

#include <stdio.h>
int main(){
    int base, exponent, i, result = 1;
    printf("Enter base and exponent \n");
    scanf("%d %d", &base, &exponent);
    for(i = 0; i < exponent; i++){
        result = result * base;
    }
    printf("%d^%d = %d", base, exponent,
result);
    return 0;
}

```

**Program 17. Check the given number is palindrome number or not using C program.**

প্রোগ্রাম এর ব্যাখ্যা : একটি সংখ্যা কে উল্টো করে পড়লেও যদি একই সংখ্যা থাকে তখন তাকে palindrome সংখ্যা বলা হবে। উদাহরণস্বরূপ ১৩১ কে উল্টা করিয়া পড়লেও একই সংখ্যা থাকে অর্থাৎ একই সংখ্যা

থাকবে ১৩১(ex: madam) প্রথমে যে সংখ্যাটি(X) ঢেক করব তার একটা কপি অন্য একটা ভারিয়েবল(Y) এ রাখবো। তারপর মূল(X) সংখ্যাটি রিভার্স করে কপি করে রাখ ভারিয়েবল (Y) এর সাথে মিলিয়ে দেবো। যদি (X) ও (Y) সমান হয় তাহলে সংখ্যাটি palindrome.

**Program:**

```

#include<stdio.h>
int main(){
    int num,r,sum=0,temp;
    printf("Enter a number: ");
    scanf("%d",&num);
    temp=num;
    while(num){
        r=num%10;
        num=num/10;
        sum=sum*10+r;
    }
    if(temp==sum)
        printf("%d is a
palindrome",temp);
    else
        printf("%d is not a
palindrome",temp);
    return 0;
}

```

**Sample output:**

Enter a number: 131  
131 is a palindrome

**Program 18. Write a C program to check given number is strong number or not.**

**Strong number :** A number is called strong number if sum of the factorial of its digit is equal to number itself. For example: 145 since  $1! + 4! + 5! = 1 + 24 + 120 = 145$

**Program:**

```

#include<stdio.h>
int main(){
    int num,i,f,r,sum=0,temp;
    printf("Enter a number: ");
    scanf("%d",&num);
    temp=num;
    while(num){
        i=1,f=1;
        r=num%10;
        while(i<=r){
            f = f*i;
            i++;
        }
        sum=sum+f;
    }
}

```

## Structure Programming (C Program)

```

    num=num/10;
}
if(sum==temp)
    printf("%d is a strong
number",temp);
else
    printf("%d is not a strong
number",temp);
return 0;
}

```

**Sample output:**

Enter a number: 145  
145 is a strong number

**Program 19. Write a c program to reverse a given number.[BUET IICT-2017, BD Computer Council - AP-2018]**

```

#include<stdio.h>
int main(){
    int num,r, rev_num =0;
    printf("Enter any number: ");
    scanf("%d",&num);
    while(num){
        r=num%10;
        rev_num = rev_num *10+r;
        num=num/10;
    }
    printf("Reversed of number: %d",
rev_num);
    return 0;
}

```

**Sample output:**

Enter any number: 4562  
Reversed of number: 2654

একটি নাম্বার রিভার্স করার প্রয়োগ এর ব্যাখ্যা :  
ধরি num = 4562 রিভার্স করে পাবো rev\_num = 2654  
এখন প্রথমে rev\_num = 0 (num কে দশ দিয়ে ভাগ করলে যে ভাগশেষ পাওয়া যাবে তার সাথে rev\_num কে দশ দিয়ে ঘণ করে rev\_num এ রাখা হবে)

শুধু এর ভিতরে ১ স্টেপ এ :  
rev\_num = rev\_num \*10 + num%10 = 2  
num = num/10 = 456

শুধু এর ভিতরে ২ স্টেপ এ :  
rev\_num = rev\_num \*10 + num%10 = 20 + 6 = 26  
num = num/10 = 45

শুধু এর ভিতরে ৩ স্টেপ এ :

rev\_num = rev\_num \*10 + num%10 = 260 + 5 = 265  
num = num/10 = 4

শুধু এর ভিতরে ৪ স্টেপ এ :  
rev\_num = rev\_num \*10 + num%10 = 265 + 4 = 2654 num = num/10 = 0

**Program 20. Write a program to find out the number of occurrence of a digit in a number.**

[DBBL -Probationary Officer (Software) -2018]

**Program:**

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int main(){
    int n, m, rem=0, p=0;
    printf("Enter your number : ");
    scanf("%d", &n);
    printf("Enter digit which you want to
find in a number : ");
    scanf("%d", &m);
    while(n!=0) {
        rem=n%10;
        if(m==rem)
            p++;
        n=n/10;
    }
    printf("Number of occurrence of a digit
in a number = %d", p);
    return 0;
}

```

**Program 21. Write down the sudo code that accepts I,n is interger and value as input, store all n integers in an array called pairs and return all pairs where the summation of individuals pair=value.[একটি আরেতে কিছু ভালু ইনপুট নিতে হবে, এখন ২ টা ভালু যোগ করে পেয়ার (জোড়া) মিলাতে হবে একটি রেফারেন্স ভালুর সাপেক্ষে][sonali and Janata bank senior officer ICT 2018]**

Input	Output
5	(1,5},{7,-1}
1 7 -1 5 -7	
6	

**Program:**

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
int main(){
    int n;
}

```

```

printf("Enter array size : ");
scanf("%d", &n);
int i=0, a[n], j=0;
printf("Enter array element : ");
for(i=0; i<n; i++) {
    scanf("%d", &a[i]);
}
printf("Enter compare number : ");
int cmp_number=0;
scanf("%d", &cmp_number);
printf("\nOutput : ");
for(i=0; i<n-1; i++) {
    for(j=i+1; j<n; j++) {
        if((a[i]+a[j])==cmp_number)
        {
            printf("{%d %d} ", a[i], a[j]);
        }
    }
}
return 0;
}

```

**Program 22.** Write a C program to convert binary to decimal.

**Program:**

```

#include<stdio.h>
int main(){
    int bn,n=0,j=1,rem,bn1;
    printf("Enter any binary number : ");
    scanf("%d",&bn);
    bn1=bn;
    while(bn!=0){
        rem=bn%10;
        n=n+rem*j;
        j=j*2;
        bn=bn/10;
    }
    printf("\nThe decimal equivalent value of
binary %d is: %d",bn1,n);
    return 0;
}

```

**Sample output:**

Enter any binary number : 101010  
The decimal equivalent value of binary 101010 is: 42

**Program 23.** C code to count the total number of digit using for loop.

**Program:**

```

#include<stdio.h>
int main(){
    int num,count=0;
    printf("Enter a number: ");
    scanf("%d",&num);
}

```

```

while(num){
    num = num/10;
    count++;
}
printf("Total digits is: %d",count);
return 0;
}

```

**Sample output:**

Enter a number: 456  
Total digits is: 3

**Program 24.** Write a C program to check prime number. [BNPS-2015, BD Land Port Authority - AP -2018]

**Program:**

```

#include <stdio.h>
int main(){
    int i, num, p = 0;
    printf("Please enter a number: \n");
    scanf("%d", &num);
    for(i=2; i<=num/2; i++){
        if(num%i==0){
            p=1;
            break;
        }
    }
    if(p==0)
        printf("%d prime number.",num);
    else
        printf("%d not a prime
number.",num);
    return 0;
}

```

**Program 25.** C program for prime numbers between 1 to N (Prime series) [Ministry of Edu:-2016, BPDB-2004]

**Program:**

```

#include<stdio.h>
int main(){
    int N, i, j, P, n;
    scanf("%d",&N);
    for(i = 2; i <= N; i++){
        P = 0;
        for(j = 2; j <= i/2; j++){
            if(i % j == 0){
                P= 1;
                break;
            }
        }
        if(P==0)
            printf("%d Prime\n",i);
    }
    return 0;
}

```

**Program 26.** A prime number is a number that is evenly divided by only 1 and itself. Write a program to your favorite language to print the first 100 prime numbers. [BD Comptition Commission- AP -2019, BD Computer Council 4 tier Data Center - AP -2019, BREB - ASE -2019]

**Program:**

```

#include<stdio.h>
int main(){
    int n, count=1, flag, i=2, j;
    printf("Enter how many prime num? :
");
    scanf("%d", &n);
    while(count <= n){
        flag = 0;
        for(j=2; j <= i/2; j++){
            if(i%j==0){
                flag=1;
                break;
            }
        }
        if(flag==0){
            printf("%d\t",i);
            count++;
        }
        i++;
    }
    return 0;
}

```

**Program 27.** Write a program to generate the  $n^{\text{th}}$  Fibonacci series in c. [Bangladesh Bank-2014, Bangladesh Bank-2016]

**Program:**

```

#include <stdio.h>
int main(){
    int i, n, a = 0, b = 1, c;
    printf("Enter how many number you want
to see");
    scanf("%d", &n);
    for (i = 1; i <= n-2; i++) {
        c = a + b;
        printf("%d ", c);
        a = b;
        b = c;
    }
    return 0;
}

```

**Sample Output:**  
0 1 1 2 3

**Program 28.** Write a program that takes a positive integer N as input and find out the sum of the digits repeatedly until the result is converged to a single digit: Sample Input: N=254189 , Sample Output: Step 1: 29, step 2: 11, step 3: 2; [Education Ministry - AP - 2014]

**Program:**

```

#include<stdio.h>
#include<math.h>
int main(){
    int n, p=0;
    printf("Enter a positive integer : ");
    scanf("%d", &n);
    while(n>9){
        int rem=0, sum=0;
        while(n){
            rem=n%10;
            sum+=rem;
            n=n/10;
        }
        p++;
        printf("Step %d : %d\n", p, sum);
        n=sum;
    }
    return 0;
}

```

**Program 29.** Write a c program to print Floyd's triangle.

**Program:**

```

#include<stdio.h>
int main(){
    int i,j,r,k=1;
    printf("Enter the range: ");
    scanf("%d",&r);
    printf("FLOYD'S TRIANGLE\n\n");
    for(i=1;i<=r;i++){
        for(j=1;j<=i;j++,k++){
            printf(" %d",k);
        }
        printf("\n");
    }
    return 0;
}

```

**Sample output:**  
Enter the range: 5  
FLOYD'S TRIANGLE  
1  
2 3

## Structure Programming (C Program)

4 5 6  
7 8 9 10  
11 12 13 14 15

**Program 30.** নিচের আউটপুট দেখাবের জন্য প্রয়োজন লিখ।  
1 2 3 6 9 18 27 54 81 162

**Program:**

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main(){
    int a=1,b=2,i,n=10;
    printf("%d %d ",a, b);
    for(i=3;i<=n-2;i++) {
        if(i%2==1){
            a=a*3;
            printf("%d ",a);
        }
        else{
            b=b*3;
            printf("%d ",b);
        }
    }
    return 0;
}
```

**Program 31.** C program to get the marks of student and print the average marks. [BD কর্তৃপক্ষ নিকাম (প্রোবশি) প্রীজ-২০১৯], Probashi Kallyan Bank – Programmer – 2019]

**Program:**

```
#include<stdio.h>
int main(){
    int n,i;
    float marks, sum = 0;
    printf("Enter any total numbers of student: ");
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        printf("Enter marks of student %d: ",i);
        scanf("%f",&marks);
        sum = sum + marks;
    }
    printf("Average marks: %f",sum/n);
    return 0;
}
```

### Sample output:

```
Enter any total numbers of student: 4
Enter marks of student 1: 85
Enter marks of student 2: 96
Enter marks of student 3: 87
Enter marks of student 4: 40
Average marks: 77.000000
```

**Program 32.** Write a program using c that reads five numbers from keyboard and display the smaller, larger and average of those numbers. [Bd development bank -senior officer it- 2017, Jagannath University - MSc Admission Test -2019]

### Program:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main(){
    int a[5];
    printf("Enter your 5 number : \n");
    for(int i=0; i<5; i++) {
        scanf("%d ", &a[i]);
    }
    int minm=a[0], maxm=a[0], sum=0;
    for(int i=0; i<5; i++) {
        if(minm>=a[i])
            minm=a[i];
        if(maxm<=a[i])
            maxm=a[i];
        sum+=a[i];
    }
    printf("Minimum number = %d", minm);
    printf("\nMaximum number = %d", maxm);
    float avg=sum/5;
    printf("\nAverage = %f", avg);
    return 0;
}
```

**Program 33.** A program to print pattern . [Land Record and Survey Department – AP -2019]

```
1
12
123
1234
12345
```

### Program:

```
#include <stdio.h>
int main(){
    int i, j;
    for(i=1;i<=5;i++) {
        for(j=1;j<=i;j++) {
            printf("%d",j);
        }
        printf("\n");
    }
    return 0;
}
```

২২ নং প্রয়োজনের ব্যাখ্যা : ধরি, n এর মান 5। প্রথম সূপে i এর মান 1 থেকে তরু এবং সূপটি ঢলবে i মান যতক্ষণ n এর চেয়ে ছোট বা সমান এবং প্রতিবার i মান এক করে বাড়বে বিভীষণ সূপে j এর মান 1

## Structure Programming (C Program)

থেকে তরু এবং সূপটি ঢলবে j মান যতক্ষণ i এর চেয়ে ছোট বা সমান এবং প্রতিবার মান 1 করে বাড়বে।

### স্টেপ ১ :

i এর মান এক এবং j এর মান এক, সূতরাং j এর মান (1) প্রিণ্ট হবে। এর পর j এর মান এক বেড়ে 2 হবে ফলে কভিশন মিথ্যা এবং সূপ থেকে বের হয়ে আসবে। সূপ এর নিচে আবার নিউ লাইন প্রিন্ট কর্মান্ত পাবে ফলে পরের পরের লাইন এ যাবে।

### স্টেপ ২ :

i এর মান বেড়ে 2 হবে এবং j এর মান 1 থেকে তরু হবে। যেহেতু কভিশন সত্য তাই j এর মান দুইবার প্রিন্ট হবে প্রথম বার 1 এবং পরের বার 2 এর পর কভিশন মিথ্যা ফলে সূপ থেকে বের হয়ে আসবে। সূপ এর নিচে আবার নিউ লাইন প্রিন্ট কর্মান্ত পাবে ফলে পরের পরের লাইন এ যাবে।

### স্টেপ ৩ :

i এর মান বেড়ে 3 হবে এবং j এর মান 1 থেকে তরু হবে। যেহেতু কভিশন সত্য তাই j এর মান তিনিবার প্রিন্ট হবে প্রথম বার 1 এবং পরের বার 2 এবং পরের বার 3। এরপর কভিশন মিথ্যা ফলে সূপ থেকে বের হয়ে আসবে। সূপ এর নিচে আবার নিউ লাইন প্রিন্ট কর্মান্ত পাবে ফলে পরের পরের লাইন এ যাবে।

### স্টেপ ৪ :

i এর মান বেড়ে 4 হবে এবং j এর মান 1 থেকে তরু হবে। যেহেতু কভিশন সত্য তাই j এর মান চারবার প্রিন্ট হবে প্রথম বার 1 এবং পরের বার 2 এবং পরের বার 3 এবং পরের বার 4। এর পর কভিশন মিথ্যা ফলে সূপ থেকে বের হয়ে আসবে। সূপ এর নিচে আবার নিউ লাইন প্রিন্ট কর্মান্ত পাবে ফলে পরের পরের লাইন এ যাবে।

### স্টেপ ৫ :

i এর মান বেড়ে 5 হবে এবং j এর মান 1 থেকে তরু হবে। যেহেতু কভিশন সত্য তাই j এর মান পাঁচবার প্রিন্ট হবে প্রথম বার 1 এবং পরের বার 2 এবং পরের বার 3 এবং পরের বার 4 এবং পরের বার 5। এর পর কভিশন মিথ্যা ফলে সূপ থেকে বের হয়ে আসবে। সূপ এর নিচে আবার নিউ লাইন প্রিন্ট কর্মান্ত পাবে ফলে পরের পরের লাইন এ যাবে।

### স্টেপ ৫ এর শেষে :

কভিশন মিথ্যা ফলে প্রথম সূপ থেকেও বের হয়ে আসবে।

**Program 34.** Program to Print Pattern.

```
54321
5432
543
54
5
```

### Program:

```
#include <stdio.h>
int main(){
    int i, j;
    for(i=1;i<=5;i++) {
        for(j=1;j<=i;j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

```
for(j=5;j>=i;j--) {
    printf("%d",j);
}
printf("\n");
}
return 0;
}
```

**Program 35.** A program to print pattern. [BD Computer Council 4 tier Data Center - AP -2019]

```
12345
1234
123
12
1
```

### Program:

```
#include <stdio.h>
int main(){
    int i, j;
    for(i=5;i>=1;i--){
        for(j=1;j<=i;j++){
            printf("%d",j);
        }
        printf("\n");
    }
    return 0;
}
```

**Program 36.** A program to print pattern.

```
*
**
***
****
*****

```

### Program:

```
#include <stdio.h>
int main(){
    int i, j;
    for(i=1;i<=5;i++){
        for(j=1;j<=i;j++){
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

**Program 37.** A program to print pattern.

```
*
**
***
*****
*****

```

### Structure Programming (C Program)

```
Program:
#include <stdio.h>
int main(){
    int i, j, k;
    for(i=1;i<=5;i++){
        for(j=i;j<5;j++){
            printf(" ");
        }
        for(k=1;k<(i*2);k++){
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

**Program 38.** Write a program to print a diamond in the following pattern.

```
*
 ***
 ****
 ***
 *
Program:
#include<iostream>
using namespace std;
int main(){
    int n, i, j, m=0;
    printf("Enter number of row : ");
    scanf("%d", &n);
    for(i=1; i<=n; i++) {
        m=(2*i)-1;
        for(j=1; j<=m; j++)
            printf("*");
        printf("\n");
    }
    for(i=n-1; i>0; i--) {
        m=(2*i)-1;
        for(j=1; j<=m; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}
```

**Program 39.** C program to find GCD/HCF and LCM of two numbers. [PSC-2009, NTRCA-2012,18, NWPGCL - AP-2019]

```
Program:
#include<stdio.h>
int main(){
    int i,a,b,gcd,lcm;
```

```
scanf("%d%d", &a, &b);
for(i = 1; i<=a&&i<=b; i++){
    if(a%i == 0 && b%i == 0){
        gcd = i;
    }
}
lcm = (a*b)/gcd;
printf("GCD: %d", gcd);
printf("LCM: %d", lcm);
return 0;
}
```

#### Program Output

```
48 180
GCD: 12
LCM: 720
```

**Program 40.** Write a program that reads (n) numbers from keyboard and display the maximum and minimum of those numbers.

```
Program:
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int i, num, max, min, total_num;
    cin>>total_num;
    cin>>num;
    max = min = num;
    for(i=1; i<total_num; i++){
        cin>>num;
        if(num>max)
            max = num;
        else if(num < min)
            min = num;
    }
    cout<<"Maximum Num: "<<
    max<<endl;
    cout<<"Minimum Num: "<<
    min<<endl;
    return 0;
}
```

**Program 41.** C program to print upper triangular matrix

```
Program:
#include <stdio.h>
#include <conio.h>
int main(){
    int rows, cols, size, row, col, sum=0;
    int inputMatrix[50][50];
```

### Structure Programming (C Program)

```
printf("Enter size square matrix\n");
scanf("%d", &size);
rows = cols = size;
printf("Enter Matrix of size %dX%d\n", rows, cols);
for(row = 0; row < rows; row++){
    for(col = 0; col < cols; col++){
        scanf("%d", &inputMatrix[row][col]);
    }
}
for(row = 0; row < rows; row++){
    for(col = 0; col < cols; col++){
        if(row < col){
            sum += inputMatrix[row][col];
        }
    }
}
```

```
printf("Sum of Upper triangular Matrix
Elements\n%d", sum);
getch();
return 0;
}
```

#### Output

```
Enter size square matrix
3
Enter Matrix of size 3X3
1 2 3
4 5 6
7 8 9
Sum of Upper triangular Matrix Elements
11
```

**Program 42.** What will be the output of the following program? Justify your answer. (Hints: incase of address write either address of x or address of y). [PGCB-2016].

```
Main () {
    int x = 3;
    int *y;
    y = &x;
    printf ("%d", y);
    printf ("%u", &y);
    printf ("%d", x);
    printf ("%u", &x);
}
```

**Output:** 1002 2010 3 1002.

**Program 43.** Write a program in C to calculate the sum of the series:  $[1 + (1+2) + (1+2+3) + \dots + (1+2+\dots+n)]$

[Multiple Ministry-2017]

**Program:**
#include <stdio.h>

```
int main(){
    int sum=0,n,i,j;
    printf("Enter the value of n:");
    scanf("%d", &n);
    for(i=1;i<n;i++){
        for(j=1;j<=i;j++){
            sum+=j;
        }
    }
    printf("%d\n",sum);
    return 0;
}
Input: 3
Output: 10
```

**Program 44.** Write a program to evaluate the series:  $1*3+2*5+3*7+\dots+n*(2n+1)$ . [DBBL - Probationary Officer (Software) -2018]

```
Program:
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main(){
    int n, sum=0, i;
    printf("Enter Nth position value: ");
    scanf("%d", &n);
    for(i=1; i<=n; i++) {
        sum+=(i*(2*i+1));
    }
    printf("Summation is %d.", sum);
    return 0;
}
```

#### Array

**Program 1.** C program to print the all elements of any array.

```
#include<stdio.h>
int main(){
    int arr[]={1,2,34,56,89};
    int i;
    for(i=0;i<5;i++){
        printf("%d ",arr[i]);
    }
    return 0;
}
```

#### Sample Output:

```
1 2 34 56 89
```

**Program 2.** C program to find the largest element in an array.

```
Program:
#include <stdio.h>
int main(){
    int a[50], size, i, big;
    scanf("%d", &size);
```

### Structure Programming (C Program)

```

        small=a[i];
    }
    printf("Smallest element: %d",small);
    return 0;
}

```

**Program 3.** Write down a function to compute the sum of the row an  $n \times m$  matrix of integer. [বাংলাদেশ সরকারি কর্ম কমিশন - সহকারী প্রত্নামা-২০১৯]

```

void matrix(int m, int n){
    int a[m][n], i, j, sum;
    printf("Enter %d x %d matrix
elements: ", m, n);
    for(i=0; i<m; i++) {
        for(j=0; j<n; j++) {
            scanf("%d", &a[i][j]);
        }
    for(i=0; i<m; i++) {
        for(j=0; j<n; j++)
            sum+=a[i][j];
        printf("Sum of row %d =
%d\n", i+1, sum);
        sum=0;
    }
}

```

**Program 4.** C code to find largest and smallest number in an array.

```

Program:
#include<stdio.h>
int main(){
    int a[50], size, i, big, small;
    printf("\nEnter the size of the array:
");
    scanf("%d", &size);
    printf("\nEnter %d elements in to the
array: ", size);
    for(i=0; i<size; i++)
        scanf("%d", &a[i]);
    big=a[0];
    for(i=1; i<size; i++){
        if(big<a[i])
            big=a[i];
    }
    printf("Largest element: %d",big);
    small=a[0];
    for(i=1; i<size; i++){
        if(small>a[i])

```

**Program 5.** A Bubble sort C program. Or A program sorts an array of integer. Write down the code that tests the sorting algorithm of written in a program. [NTRCA-2012, Combined Five Bank - AME - 2019, Combined 3 banks, Combined (BHBFC & BKB) Assis. Programmer 2018, BD Development Bank - SO -2017]

বা, তোমার ফ্লাইনের জাতীয় sort করার জন্য একটি C Program লিখুন। এখানে কোন Built-in function ব্যবহার করা যাবে না। [Food ministry-2021]

**Program:**

```

#include <stdio.h>
int main(){
    int data[100], i, n, step, temp;
    printf("Enter the number of elements
to be sorted: ");
    scanf("%d", &n);
    for(i=0; i<n; ++i){
        printf("%d. Enter element: ", i+1);
        scanf("%d", &data[i]);
    }
    for(step=0; step<n-1; ++step)
    for(i=0; i<n-step-1; ++i){
        if(data[i]>data[i+1]){
            temp=data[i];
            data[i]=data[i+1];
            data[i+1]=temp;
        }
    }
    printf("In ascending order: ");
    for(i=0; i<n; ++i)
        printf("%d ", data[i]);
    return 0;
}

```

**Program 6.** C Program to Sort set of strings in alphabetical order .[NTRCA-2010, বাংলাদেশ সরকারি কর্ম কমিশন - সহকারী প্রত্নামা-২০১৯]

```

Program:
#include<stdio.h>
#include<string.h>
int main(){
    int i, j, count;
    char str[25][25], temp[25];
    puts("How many strings u are going to
enter? ");

```

### Structure Programming (C Program)

```

scanf("%d", &count);

puts("Enter Strings one by one: ");
for(i=0; i<=count; i++)
    gets(str[i]);
for(i=0; i<=count; i++){
    for(j=i+1; j<=count; j++){
        if(strcmp(str[i], str[j])>0){
            strcpy(temp, str[i]);
            strcpy(str[i], str[j]);
            strcpy(str[j], temp);
        }
    }
}
printf("Order of Sorted Strings:");
for(i=0; i<=count; i++)
    puts(str[i]);

return 0;
}

```

**Program 7.** Write a program to sort a list of element using the insertion sort algorithm. [Investment Corporation of BD - AP -2019]

**Program:**

```

#include<stdio.h>
#include<string.h>
void insertionSort(int arr[], int
n){
    int i, key, j;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] >
key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

void printArray(int arr[], int n){
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

int main(){
    int arr[] = { 12, 11, 13, 5, 6
};
    int n = sizeof(arr) /
sizeof(arr[0]);
    insertionSort(arr, n);
    printArray(arr, n);
}

```

**Program 8.** Write a program using c/c++ that read five number from keyboard and display the Smaller,Larger, and Average of these number.

[ এ টি সংখ্যা কী-বোর্ড থেকে ইনপুট নিয়ে বড়, ছোট, আভাবেজ বের করার জন্য একটি সি/সি++ এ প্রোগ্ৰাম লিখ?][Senior IT Officer BDB-2017]

**Program:**

```

#include<stdio.h>
int main(){
    int a[5], i, j, temp;
    float avg=0;
    for(i=0; i<5; i++){
        printf("Enter %d st
value: ", (i+1));
        scanf("%d", &a[i]);
    }
}

```

for(i=0; i<5; i++){
 for(j=0; j<5-i; j++){
 if(a[i]>a[i+1]){
 temp=a[i];
 a[i]=a[i+1];
 a[i+1]=temp;
 }
 }
 avg+=a[i];
}
printf("The large value is=%d\n", a[0]);
printf("The small value is=%d\n", a[5-1]);
printf("The average value is=%f\n", avg/5);
return 0;
}

#### Output:

```

Enter 1st value:3
Enter 2nd value:9
Enter 3rd value:32
Enter 4th value:28
Enter 5th value:7

```

```

The large value is= 32
The small value is= 3
The average value is= 15.800000

```

**Program 9.** Write a c program for swapping of two arrays.[দুইটি আরের ভালু পরম্পরাগত পরিবর্তনের জন্য একটি সি প্রোগ্ৰাম লিখ?]

**Program:**

```

#include<stdio.h>

```

## Structure Programming (C Program)

```

int main(){
    int a[10],b[10],c[10],i;
    printf("Enter First array->");
    for(i=0;i<10;i++)
        scanf("%d",&a[i]);
    printf("\nEnter Second array->");
    for(i=0;i<10;i++)
        scanf("%d",&b[i]);
    for(i=0;i<10;i++){
        c[i]=a[i];
        a[i]=b[i];
        b[i]=c[i];
    }
    printf("\nArrays after swapping");
    printf("\nFirst array->");
    for(i=0;i<10;i++){
        printf("%d",a[i]);
    }
    printf("\nSecond array->");
    for(i=0;i<10;i++){
        printf("%d",b[i]);
    }
    return 0;
}

```

**Sample Output:**

```
Enter First array->4 5 8 9 5 4 6 5 2 2
Enter Second array->5 8 5 5 5 5 8 8 9 4
Arrays after swapping
First array->5855558894
Second array->4589546522
```

**Program 10.** Write a program to find out the second max element in an array without using library function. [ লাইব্রেরী ফাংশন ব্যবহার না করে দিতী সর্বোচ্চ উপাদান বের করার প্রয়োগ শিখ ] [DBBL -Probationary Officer (Software) -2018]

## Program

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main(){
    int n;
    printf("Enter array size : ");
    scanf("%d", &n);
    int i, j, arr[n], temp=0;
    printf("Enter array element: ")
    for(i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }
    for(i=0; i<n; i++){
        for(j=i+1; j<n; j++) {
            if(arr[i]>arr[j]){
                temp=arr[j];
                arr[j]=arr[i];
                arr[i]=temp;
            }
        }
    }
    printf("Sorted array elements are: ");
    for(i=0; i<n; i++)
        printf("%d ", arr[i]);
    getch();
}
```

```
        arr[j]=arr[i];
        arr[i]=temp;
    }
}
printf("Second maximum number: %d"
arr[n-2]);
return 0;
}
```

**Program 11.** Suppose an array is {4, 5, 6, 7}. Write a program that will be output like {{(4,5), (4,6), (4,7), (5,6), (5,7), (6,7)}}. [North-West Power generation Company Ltd. - AE -2018]

## Program

```
#include<stdio.h>
int main(){
    int n;
    printf("Enter amount of array elements");
    scanf("%d", &n);
    int arr[n], i, j;
    printf("Enter %d elements : ");
    for(i=0; i<n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Output:\n{");
    for(i=0; i<n-1; i++) {
        for(j=i+1; j<n; j++) {
            printf("(%d, %d)", arr[i], arr[j]);
            if(i<n-2)
                printf(",");
        }
    }
    printf("}");
    return 0;
}
```

## Program 12. C Program to Multiply Two Matrices Using Multi-dimensional Arrays[মাট্রিস পরিমাণ করে এবং গুণাগুণ করে নির্বাচন করা]

## Program

```
#include<stdio.h>
#include<stdlib.h>
int main(){
int
a[10][10],b[10][10],mul[10][10],r,c,i,j,
system("cls");
printf("enter the number of row=");
scanf("%d",&r);
printf("enter the number of column=");
scanf("%d",&c);
```

# Structure Programming (C Programming)

```

printf("enter the first matrix
element=\n");
for(i=0;i<r;i++) {
    for(j=0;j<c;j++) {
        scanf("%d",&a[i][j]);
    }
}
printf("enter the second matrix
element=\n");
for(i=0;i<r;i++) {
    for(j=0;j<c;j++) {
        scanf("%d",&b[i][j]);
    }
}
printf("multiply of the matrix=\n");
for(i=0;i<r;i++) {
    for(j=0;j<c;j++) {
        mul[i][j]=0;
        for(k=0;k<c;k++) {
            mul[i][j]+=a[i][k]*b[k][j];
        }
    }
}
//for printing result
for(i=0;i<r;i++) {
    for(j=0;j<c;j++) {
        printf("%d\t",mul[i][j]);
    }
    printf("\n");
}
return 0;
}

```

$$\begin{array}{c}
 \begin{array}{ccccc}
 & & \text{→} & & \\
 \boxed{\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}} & \times & \boxed{\begin{array}{cc} 10 & 11 \\ 20 & 21 \\ 30 & 31 \end{array}}
 \end{array} \\
 = \begin{bmatrix} 1 \times 10 + 2 \times 20 + 3 \times 30 & 1 \times 11 + 2 \times 21 + 3 \times 31 \\ 4 \times 10 + 5 \times 20 + 6 \times 30 & 4 \times 11 + 5 \times 21 + 6 \times 31 \end{bmatrix} \\
 = \begin{bmatrix} 10+40+90 & 11+42+93 \\ 40+100+180 & 44+105+186 \end{bmatrix} = \begin{bmatrix} 140 & 143 \\ 320 & 333 \end{bmatrix}
 \end{array}$$

**Program 15.আলফাৰেটিক অর্ডাৰে স্ট্ৰাই সাজানোৱ জন্য একটি  
ফাঁশন লিখন। [বিভিন্ন মুদ্রণালয়-২০১৯-AP]**

```
উত্তর: Void sort(){
    int i,j,count;
    char str[25][25],temp[25];
    puts("How many strings you are going to enter?: ");
    scanf("%d",&count);
    puts("Enter Strings one by one: ");
    for(i=0;i<count;i++)
        gets(str[i]);
    for(i=0;i<count;i++)
        for(j=i+1;j<count;j++)
            if(strcmp(str[i],str[j])>0)
                swap(str[i],str[j]);
    for(i=0;i<count;i++)
        puts(str[i]);
```

```
for(i=0;i<=count;i++)
    gets(str[i]);
for(i=0;i<=count;i++)
    for(j=i+1;j<=count;j++){
        if(strcmp(str[i],str[j])>0){
            strcpy(temp,str[i]);
            strcpy(str[i],str[j]);
            strcpy(str[j],temp);
        }
    }
printf("Order of Sorted Strings:");
for(i=0;i<=count;i++)
    puts(str[i]);
```

॥ Program using Recursive function ॥

**Program 1. Count the digits of a given number C language using recursion.** [একটি সংখ্যাটে কতজনো ডিজিট আছে তা গণনা করার অন্য প্রয়োগ লিখ?]

### **Program:**

```
#include<stdio.h>
int countDigits(num);
int main(){
    int num,count;
    printf("Enter a number: ");
    scanf("%d",&num);
    count = countDigits(num);
    printf("Total digits is: %d",count)
    return 0;
}
int countDigits(int num){
    static int count=0;
    if(num!=0){
        count++;
        countDigits(num/10);
    }
    return count;
}
```

**Sample output:**  
Enter a number: 12345  
Total digits is: 7

**Program 2. Program for find out the sum of digits in c using recursion.** [রিকার্শন ফাংশন ব্যবহার করে একটি সংখ্যার ইনডিভিজুয়াল ডিজিটের যোগফল নির্ণয়ের প্রোগ্রাম লিখ? ]

Program

```
#include<stdio.h>
int getSum(int);
int main(){
    int num,sum;
    printf("Enter a number: ");
    scanf("%d",&num);
    sum = getSum(num);
    printf("Sum of digits of number
%d",sum);
    return 0;
```

## Structure Programming (C Program)

```

}
int getSum(int num){
    static int sum = 0,r;
    if(num!=0){
        r=num%10;
        sum=sum+r;
        getSum(num/10);
    }
    return sum;
}

```

### Sample output:

Enter a number: 45

Sum of digits of number: 9

**Program 3. C program to reverse a number using recursion.** [ রিকার্সন ব্যবহার করে একটি নম্বৰ রিভার্স করার প্রোগ্ৰাম লিখ? ]

### Program:

```

#include<stdio.h>
int main(){
    int num,reverse;
    printf("Enter any number: ");
    scanf("%d",&num);
    reverse=rev(num);
    printf("Reverse of number: %d",reverse);
    return 0;
}

int sum = 0, r;
int rev(int num){
    if(num){
        r=num%10;
        sum=sum*10+r;
        rev(num/10);
    }
    else
        return sum;
}

```

### Sample output:

Enter any number: 456

Reverse of number: 654

**Program 4. Factorial program by recursion in c.** [ রিকার্শন ব্যবহার করে একটি সংখ্যা ফ্যাক্টোরিয়াল বের করার প্রোগ্ৰাম লিখ? ] [Multiple Ministry-2016] [ICT D-AP-2021]

### Program:

```

#include<stdio.h>
int fact(int);
int main(){
    int num,f;
    printf("\nEnter a number: ");
    scanf("%d",&num);
    int fact (int n){

```

```

        f=fact(num);
        printf("\nFactorial of %d is:
%d",num,f);
        return 0;
    }
    int fact(int n){
        if(n==1)
            return 1;
        else
            return(n*fact(n-1));
    }

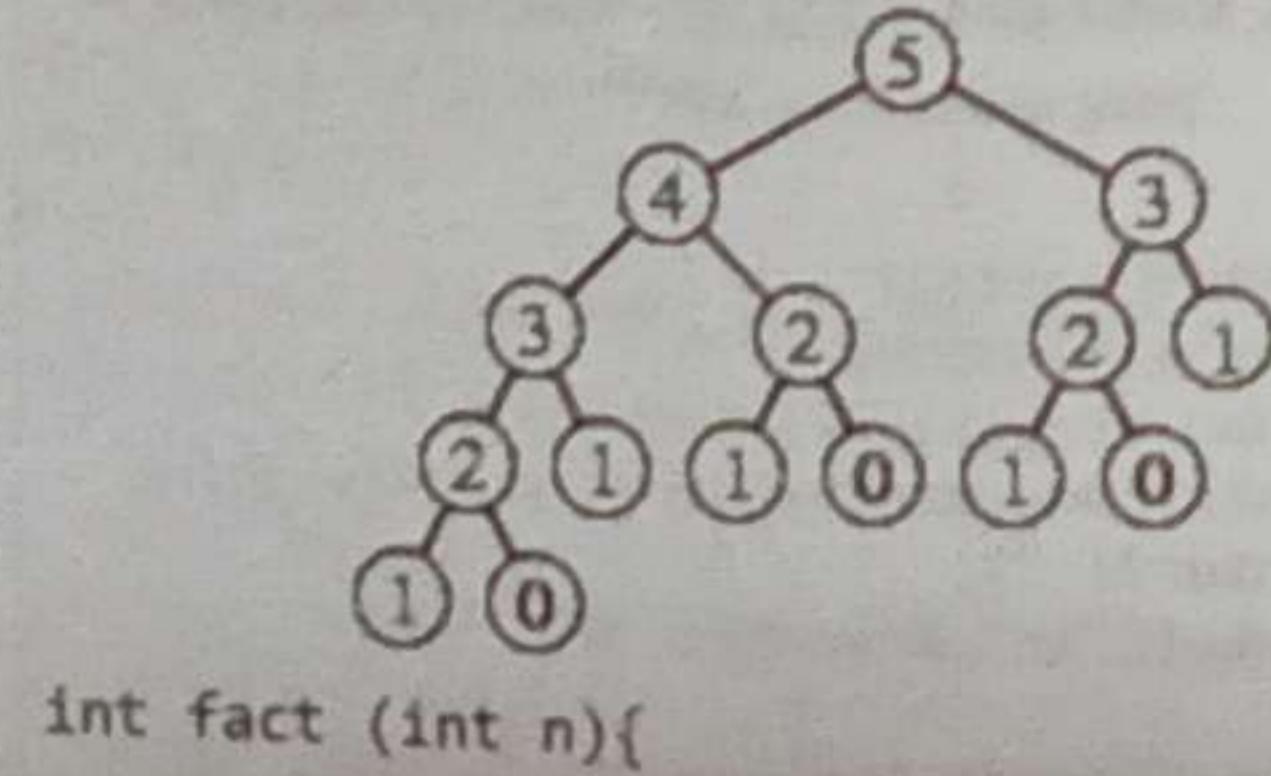
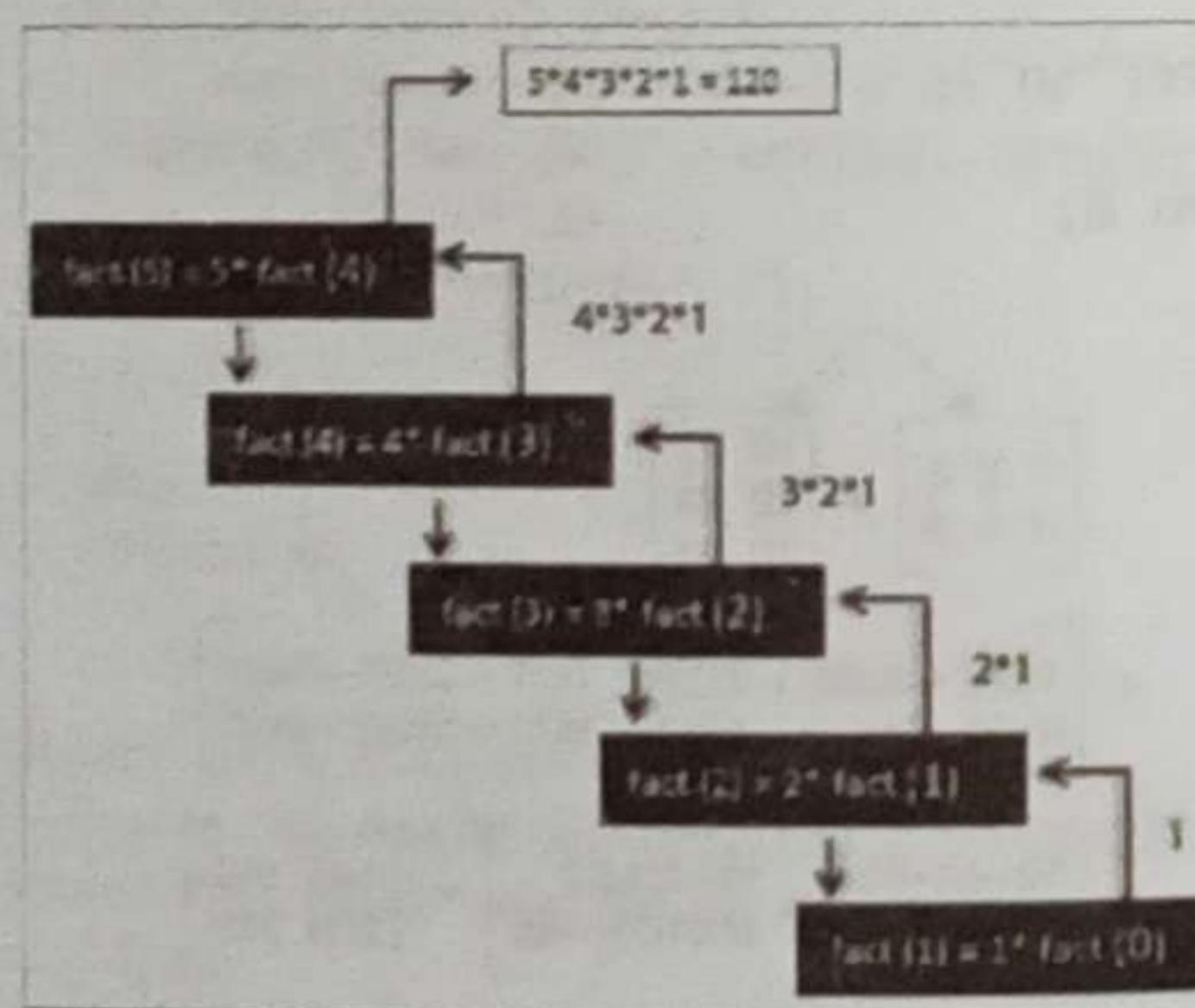
```

### Sample output:

Enter a number: 5

Factorial of 5 is: 120

৪ নং প্রোগ্ৰামৰ ব্যাখ্যা : যদি একটি ফাংশন fact () যা নিজেই নিজের একটি কল স্টেটমেন্ট অথবা দ্বিতীয় কোন ফাংশনের একটি কল স্টেটমেন্ট দ্বাৰা fact () এ কিবে আসে তখন fact () কে রিকার্শন ফাংশন বলে। ফাংশনটি যাতে অনিনিটি সময় ধৰে না চলে তাৰ জন্য কিছু বিষয় মেনে চলা হয়। এমন কিছু বৈশিষ্ট্য যাৰ জন্য ফাংশনটি আৰু নিজেকে কল কৰে না। অৰ্থাৎ ফাংশনেৰ কিছু আৰ্ডেন্ট থাকবে যাৰ জন্য ফাংশনটি আৰু নিজেকে কল কৰবে না। একে বেস ভ্যালু বলে।প্ৰত্যোক সময়ে যথন ফাংশনটি নিজেকে কল কৰে তখন ফাংশনেৰ আৰ্ডেন্ট বেস ভ্যালুৰ নিকটৰতী হচ্ছে থাকবে। অৰ্থাৎ রিকার্সন এৰ ক্ষেত্ৰে একটি শৰ্ত লাগবে যাৰ জন্য ফাংশনটি কোন এক সময় টার্মিনেট কৰবে এবং অতিবার সেই শৰ্তৰ দিকে ফাংশনটি আগবঢ়ে।আমৰা ফ্যাক্টোরিয়াল হিসাবেৰ ফাংশনটি লক কৰি।



## Structure Programming (C Program)

```

if(n==1) return 1;
return n*fact(n-1);
}

```

fact(5) এৰ মান প্ৰিণ্ট কৰা হইলে প্ৰিণ্ট কৰবে 120।  
এখনে if(n==1) return 1; এটা হল টাৰ্মিনেট কৰাৰ শৰ্ত, যাকে  
বলে বেস ভ্যালু। এখন এৰ ভিতৰ যে ফাংশনটি কল কৰা হয়েছে তা হল  
fact(n-1)। অৰ্থাৎ  
fact(5) ফাংশনটি কল কৰবে fact(4) ফাংশনটিকে  
fact(4) ফাংশনটি কল কৰবে fact(3) ফাংশনটিকে  
fact(3) ফাংশনটি কল কৰবে fact(2) ফাংশনটিকে  
fact(2) ফাংশনটি কল কৰবে fact(1) ফাংশনটিকে

যখন fact(1) ফাংশনটিৰ কাজ শৰ্ত হবে n==1 এই শৰ্তী সত্য  
হওয়াৰ জন্য ফাংশনটি ১ রিটোৰ্ন কৰবে। এই রিটোৰ্ন না কৰলে অনিনিটি  
সময় ধৰে প্ৰোগ্ৰামটি চলতো।

তাৰপৰ এই fact(1), 1 রিটোৰ্ন কৰবে fact(2) কে  
fact(2), 2fact(1) = 2\*1 = 2 রিটোৰ্ন কৰবে fact(3) কে  
fact(3), 3fact(2) = 3\*2 = 6 রিটোৰ্ন কৰবে fact(4) কে  
fact(4), 4fact(3) = 4\*6 = 24 রিটোৰ্ন কৰবে fact(5) কে  
অৱ তাই fact(5) এৰ মান হবে 5fact(4) = 5\*24 = 120।  
রিকার্সনেৰ স্বচেয়ে বড় সুবিধা হল কোড ছোট হয়। অনেক বড় কাজ  
শুৰু হোট কোভে হয়ে যায়। যেখানে স্ট্যাক ব্যবহাৰ কৰা হয় দেখাবেন  
রিকার্সন খুব ভালো কাজ কৰে। আপনি লুপেৰ সব কাজ রিকার্সন দিয়ে  
কৰাতে পাৰবেন। ডায়নামিক প্ৰোগ্ৰামিং এ রিকার্সন একটি শক্তিশালী  
টেকনিক।

**Program 5. C code to print Fibonacci series by recursion.** [Titas Gas-2018]

### Program:

```

#include<stdio.h>
int Fibonacci(int);
main(){
    int n, i = 0, c;
    scanf("%d",&n);
    printf("Fibonacci series\n");
    for ( c = 1 ; c <= n ; c++ ){
        printf("%d ", Fibonacci(i));
        i++;
    }
    return 0;
}

int Fibonacci(int n){
    if ( n == 0 )
        return 0;
    else if ( n == 1 )
        return 1;
    else
        return ( Fibonacci(n-1) +
Fibonacci(n-2) );
}

```

**Program 6. C code to check a number is prime number or not by recursion.** [NTRCA-2012]

### Program:

```

#include<stdio.h>
int isPrime(int,int);
int main(){
    int num,prime;
    printf("Enter a positive number: ");
    scanf("%d",&num);
    prime = isPrime(num,num/2);
    if(prime==1)
        printf("%d is a prime number",num);
    else
        printf("%d is not a prime number",num);
    return 0;
}

```

```

int isPrime(int num,int i){
    if(i==1){
        return 1;
    }else{
        if(num%i==0)
            return 0;
        else
            isPrime(num,i-1);
    }
}

```

### Sample output:

Enter a positive number: 13

13 is a prime number

**Program 7. Find power of a number using recursion using C program.**

### Program:

```

#include <stdio.h>
int power(int n1, int n2);
int main(){
    int base, powerRaised, result;
    printf("Enter base number: ");
    scanf("%d",&base);
    printf("Enter power number(positive integer): ");
    scanf("%d",&powerRaised);
    result = power(base, powerRaised);
    printf("%d^%d = %d", base,
powerRaised, result);
    return 0;
}

int power(int base, int powerRaised){
    if ( powerRaised != 0 )
        return base * power(base, powerRaised-1);
    else
        return 1;
}

```

```

    else
        return 1;
}

```

**Program 8. Sum of digits in c using recursion.**

**Program:**

```

#include<stdio.h>
int main(){
    int num,x;
    clrscr();
    printf("\nEnter a number: ");
    scanf("%d",&num);
    x=findsum(num);
    printf("Sum of the digits of %d is:
%d",num,x);
    return 0;
}
int findsum(int n){
    if(n){
        r=n%10;
        s=s+r;
        findsum(n/10);
    }
    else
        return s;
}

```

**Sample output:**

Enter a number: 145

Sum of the digits of 145 is: 10

**Program 9. GCD of Two Numbers using Recursion**

**Program:**

```

#include <stdio.h>
int hcf(int n1, int n2);
int main() {
    int n1, n2;
    printf("Enter two positive integers:
");
    scanf("%d %d", &n1, &n2);
    printf("G.C.D of %d and %d is %d.",
n1, n2, hcf(n1, n2));
    return 0;
}

int hcf(int n1, int n2) {
    if (n2 != 0)
        return hcf(n2, n1 % n2);
    else
        return n1;
}

```

**Program 10. Write a function sumdigits to find output like below:**

**Input=123**

**Output=6**

[BTCL-Assistant Manager (Technical)-2021]

```

#include <stdio.h>
int main(){
    int n, sum;
    printf("Enter an integer\n");
    scanf("%d", &n);
    sum = sumdigits(n);
    printf("Sum of digits of %d = %d\n", n, sum);
    return 0;
}
int sumdigits(int num)
{
    ***
}

```

**Solution:**

```

if(num==0)
    return 0;
else
    return num%10+sumdigits(num/10);

```

1. The following C program intended to calculate  $n^r$ , the  $r$ -th non-negative integer power of a non-negative number  $n$ , using recursion. You need to complete the program by writing the recursive function. Also you need to minimize the number of recursive call to make the program efficient.

[বিকাশ ব্যবহার করে  $n^r$  ক্যালকুলেট করার জন্য নিচের ফাংশনটি কমপ্টি

করতে একটি সি প্রোগ্রাম লিখতে হবে, যেখানে  $n,r$  হলো নম্বরটি নথিত নাম।]

[Titas Gas -2021-AE - CSE] [10]

```

#include<stdio.h>
int power(int n, int r) {
    ***
}
int main(){
    int n, r;
    scanf("%d%d", &n, &r);
    printf("%d", power(n,r));
    return 0;
}

```

**Ans:**

```

if (r==0) return 1;
else {return n* power (n, r-1);}

```

**String**

**Program 1. Write a program to print a string in C language.**

```

#include<stdio.h>
#include<conio.h>
int main(){
    printf("bitBox");
    return 0;
}

```

**Sample output:** bitBox

**Program 2. C program to accept a string from users and print it.**

```

#include<stdio.h>
int main(){
    char str[20];
    printf("Enter the string :");
    scanf("%s",&str);
    printf("Result:%s",str);
    return 0;
}

```

**Sample output:**

Enter the string :hello  
Result:hello

**Program 3. Write a function void ChangeCase And Reverse (Char \*s) using C programming language that will print the characters in the string S in reverse order and will also change the uppercase letters to lowercase and lowercase letters to uppercase. For example if you call changeCaseAndReverse ("Hello There"), it will print EREHtOLLEh. [Uttora Bank - SO (IT) -2019]**

```

#include<iostream>
using namespace std;
void changeCaseAndReverse(string &str){
    int ln = str.length();
    for (int i=0; i<ln; i++) {
        if (str[i]>='a' && str[i]<='z')
            str[i] = str[i] - 32;
        else if(str[i]>='A' && str[i]<='Z')
            str[i] = str[i] + 32;
    }
    for (int i = 0; i < ln / 2; i++)
        swap(str[i], str[ln - i - 1]);
}
int main(){
    string str = "engr. harun ";
    changeCaseAndReverse(str);
    cout << str;
    return 0;
}

```

**Program 4. C program to accept a string and find out total numbers of characters.**

```

#include <stdio.h>
int main(){
    char c[1000],ch;
    int i,count=0;
    printf("Enter a string: ");
    gets(c);
    printf("Enter a character to find
frequency: ");
    scanf("%c",&ch);
    for(i=0;c[i]!='\0';++i) {
        if(ch==c[i])
            ++count;
    }
    printf("Frequency of %c = %d", ch,
count);
    return 0;
}

```

**Sample output:**

Enter a string: BITBOX  
Enter a character to find frequency: B  
Frequency of B = 2

**Library Function Use করে:**

সি strlwr() ফাংশন: strlwr(string) ফাংশন একটি স্ট্রিংকে ছোট ঘাতের বর্ণে রূপান্তর করে।

```

#include<stdio.h>
#include<conio.h>
int main(){
    char str[30];
    clrscr();
    printf("Enter string: ");
    gets(str); //কনসোল থেকে স্ট্রিং পাঠ(reads) করে।
    printf("String is: %s",str);
    printf("\nLower String is: %s",strlwr(str));
    getch();
}

```

**অটোপু**

Enter string: bitBoxIT  
String is: bitBoxIT  
Lower String is: bitboxit

সিstrupr() ফাংশন:strupr(string) ফাংশন একটি স্ট্রিংকে বড় ঘাতের বর্ণে রূপান্তর করে।

```

#include<stdio.h>
#include<conio.h>
int main(){
    char str[30];
    clrscr();
    printf("Enter string: ");
    gets(str); //কনসোল থেকে স্ট্রিং পাঠ(reads) করে।
}

```

```

printf("String is: %s",str);
printf("\nUpper String is: %s",strupr(str));
getch();
}

```

Enter string: bitBoxIT  
String is: bitBoxIT  
Upper String is: BITBOXIT

**Program 5. C program to convert upper case to lower case with out library function.**

```

#include<stdio.h>
#include<string.h>
int main(){
    char str[20];
    int i;
    printf("Enter any string: ");
    scanf("%s",str);
    printf("The string is: %s",str);
    for(i=0;i<=strlen(str);i++){
        if(str[i]>=65&&str[i]<=90)
            str[i]=str[i]+32;
    }
    printf("\nThe string in Lower case is:
%s",str);
    return 0;
}

```

**Sample output:**

Enter any string: CSE14BATCH  
The string is: CSE14BATCH  
The string in Lower case is: cse14batch

**Program 6. C program to convert lower case to upper case with out library function.**

```

#include<stdio.h>
int main(){
    char str[20];
    int i;
    printf("Enter any string: ");
    scanf("%s",str);
    printf("The string is: %s",str);
    for(i=0;i<=strlen(str);i++){
        if(str[i]>=97&&str[i]<=122)
            str[i]=str[i]-32;
    }
    printf("\nThe string in uPPERcase is:
%s",str);
    return 0;
}

```

**Sample output:**

Enter any string: duetcse14batch  
The string is: duetcse14batch

The string in uPPERcase is: DUETCSE14BATCH

**Program 7. Write a program for find the length of a String.**

```

#include<stdio.h>
int main(){
    int i,j=0;
    char str[40];
    printf("Enter a string:");
    gets(str);
    int sl=strlen(str);
    printf("The length is:%d",sl);
    return 0;
}

```

**Output:**

Enter a string: DUET  
The length is: 4

**Program 8. Write a program for Reverse a String.** [Investment Corporation of BD - AP -2019]

```

#include<stdio.h>
int main()
{
    int i,j=0;
    char str[100],str_new[100];
    gets(str);
    int sl=strlen(str);
    for(i=sl-1;i>=0;i--){
        str_new[j]=str[i];
        j=j+1;
    }
    str_new[j]='\0';
    printf("%s\n",str_new);
    return 0;
}

```

**Input:** Anisha

**Output:** ahnsiA

**Library Fuction Use করে:**

দি strrev() কালেন: strrev(string) কালেন একটি স্ট্রিংকে উল্টো দেলে। অর্থাৎ স্ট্রিং এর পথম ক্যারেক্টরগুলো শেষে এবং শেষ ক্যারেক্টরগুলো পথমে আসে।

```

#include<stdio.h>
#include<conio.h>
int main(){
    char str[30];
    clrscr();
    printf("Enter string: ");
    gets(str); //কমসোল থেকে স্ট্রিং পাঠ(reads) করা
    printf("String is: %s",str);
    printf("\nReverse String is: %s",strrev(str));
}

```

getch();

**Output:**

Enter string: bitBoxIT  
String is: bitBoxIT  
Reverse String is: TIxoBtib

**Program 9. Write a program in C to count word in a string or a sentence without using library function.** [ BREB – ASE -2019, Land Record and Survey Department – AP -2019]

```

#include<stdio.h>
int main(){
    char s[100];
    int i=0, p=0, n=0;
    printf("Enter the string or a sentence
: ");
    scanf("%[^\\n]s", s);
    while(s[n]!='\0')
        n++;
    for(i=0; i<n; i++){
        if(s[i]==' ' && s[i+1]!=' ')
            p++;
    }
    printf("Number of word : %d", p+1);
    return 0;
}

```

**Program 10. Copy one string value an other without library function.**

```

#include<stdio.h>
int main() {
    char s1[100], s2[100];
    int i;
    printf("\nEnter the string :");
    gets(s1);
    i = 0;
    while (s1[i] != '\0') {
        s2[i] = s1[i];
        i++;
    }
    s2[i] = '\0';
    printf("\nCopied String is: %s ", s2);
    return (0);
}

```

**Program 11. Write a program to count all vowels present in the string in C language.**

```

#include<stdio.h>
#include<conio.h>
main(){
    int i,vowel=0;
}

```

```

char s[100]={0};
printf("Enter the string: ");
scanf("%s",s);
for(i=0;s[i]!='\0';i++){
    if((s[i]=='A')||(s[i]=='E')||(s[i]=='O')||
    (s[i]=='U')||(s[i]=='I')||(s[i]=='a')||(s[i]=='e')||(s[i]=='o')||(s[i]=='u')||(s[i]=='i'))
        vowel++;
}
printf("\nVowels : %d",vowel);
}

```

**Sample output:**

Enter the string: bitbox  
Vowels : 2

**Program 12. Swapping of strings using C programming language** [NTRCA-2016]

```

#include<stdio.h>
int main(){
    int i=0,j=0,k=0;
    char str1[20],str2[20],temp[20];
    puts("Enter first string");
    gets(str1);
    puts("Enter second string");
    gets(str2);
    while(str1[i]!='\0'){
        temp[j++]=str1[i++];
    }
    temp[j]='\0';
    i=0,j=0;
}

```

```

while(str2[i]!='\0'){
    str1[j++]=str2[i++];
}
str1[j]='\0';
i=0,j=0;
while(temp[i]!='\0'){
    str2[j++]=temp[i++];
}

```

```

str2[j]='\0';
printf("After swaping the strings
are\n");
puts(str1);
puts(str2);
return 0;
}

```

**Sample output:**

Enter first string :RKB  
Enter second string :HARUN  
After swaping the strings are  
HARUN  
RKB

### Structure Programming (C Program)

**Program 13.** Write a C program to check given string is palindrome number or not. [DBBL-2016, Land Record and Survey Department - AP-2019]

```
#include<string.h>
#include<stdio.h>
int main(){
    char str[100]={0},rev[100]={0};
    int i,j;
    printf("\nEnter a string:");
    scanf("%s",str);
    for(i=strlen(str)-1,j=0;i>=0;i--,j++)
        rev[j]=str[i];
    if(strcmp(rev,str))
        printf("\nThe string is not a palindrome");
    else
        printf("\nThe string is a palindrome");
    return 0;
}
```

#### Sample output:

```
Enter a string:MADAM
The string is a palindrome
```

**[HW:** Compute the sum of palindrome numbers among 100 to 999 (like 101, 313 etc). [BUET MSc Admission Test -2019]

**Program 14.** Write a program to remove spaces and converted the words into Camel Case from a given sentence.

```
#include <bits/stdc++.h>
using namespace std;
string convert(string s){
    int n = s.length();
    int res_ind = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] == ' ')
            continue;
        else
            s[res_ind++] = s[i];
    }
    return s.substr(0, res_ind);
}
int main(){
    string str = "bitBox it job
solution";
    cout << convert(str);
}
```

return 0;

**Input:** bitBox it job  
**Output:** bitBoxItJobSolution

সি strcpy() ফাংশন: strcpy(destination, source) কাছে source স্ট্রিং থেকে স্ট্রিংকে destination এ কপি করে।

উদাহরণ: নিচের উদাহরণে strcpy() ফাংশনের মাধ্যমে ch1[30]

স্ট্রিংকে ch2[30] স্ট্রিং এর মধ্যে কপি করে দেখানো হলো।

```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
    char ch1[30]={ 'b', 'i', 't', ' ', 'B', 'o', 'x', 'i', 't', 'j', 'o', ' ', '0' };
    char ch2[30];
    strcpy(ch2,ch1);
    printf("The value of second string is: %s",ch2);
}
```

#### আউটপুট

The value of second string is: bitBoxitjob

সি strcat() ফাংশন: strcat(first\_string, second\_string) ফাংশন দুটি স্ট্রিংকে একসঙ্গে যোগ(concatination) করে প্রথম স্ট্রিং হিসাবে রিটার্ন করে।

```
#include <stdio.h>
int main(){
    char ch1[10] = { 'h', 'e', 'l', 'l', 'o', ' ', '0' };
    char ch2[10] = { 'b', 'i', 't', ' ', 'B', 'o', 'x', 'i', 't', 'j', 'o', ' ', '0' };
    strcat(ch1,ch2);
    printf("The value of first string is: %s",ch1);
}
```

#### আউটপুট

The value of first string is: hellbitBox

সি strcmp() ফাংশন: strcmp(first\_string, second\_string) ফাংশন দুটি স্ট্রিং এর মধ্যে তুলনা করে এবং উভয় স্ট্রিং সমান হলে 0(শূণ্য) রিটার্ন করে। কনসোল থেকে স্ট্রিং পাঠ(read) করার জন্য এখনে আমরা gets() ফাংশন ব্যবহার করেছি।

```
#include <stdio.h>
int main(){
    char str1[30],str2[30];
    printf("Enter first string: ");
    gets(str1); // কনসোল থেকে স্ট্রিং পাঠ(read) করা
    printf("Enter second string: ");
    gets(str2);
    if(strcmp(str1,str2)==0)
        printf("Two strings are equal");
    else
        printf("Two strings are not equal");
}
```

### Structure Programming (C Program)

#### আউটপুট

Enter first string: bitBox  
Enter second string: IT  
Two strings are not equal

#### File

**প্রোগ্রাম 1.** কি-বোর্ড থেকে ইনপুট নিয়ে আউটপুট ফাইল এ রাখার প্রোগ্রাম লেখ। (print numbers from 1 to N.)

```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
    int i,n;
    scanf("%d",&n);
    freopen("output.txt","w",stdout);
    for(i=1;i<=n;i++){
        printf("%d ",i);
    }
    return 0;
}
```

**Output:** 2686744(Address of the variable).

**প্রোগ্রাম e.** A file called myfile.txt contains a list of days and power shortage of that day like this ....

1,4357  
2,34554 .....

Write a C program to find the power shortage of the 30th day. [BPDB-2018]

**Ans:**

```
#include <iostream>
#include <fstream>
int main(){
    int a,i;
    freopen("myfile.txt","r",stdin);
    scanf("%d",&a);
    for(i=0;i<30;i++){
        scanf("%d",&a);
        printf("%d,%d\n", (i+1),a);
    }
    return 0;
}
```

#### Output Program

**প্রোগ্রাম 3.** নিচে programming C এর একটি অংশ দেয়া হল, অংশটি থেকে Output এর মান কি কি হবে বের কর।

```
int main(){
    int a,b,c,d;
    a=15;
    b=10;
    c=++a - b;
    printf("a=%d b=%d c=%d\n",a,b,c);
    d=b++ + a;
    printf("a=%d b=%d d=%d\n",a,b,d);
    while(num>=1){
        f=f*num;
        num--;
    }
    printf("Factorial value is: %d",f);
    return 0;
}
```

**Output:**

```
a=16 b=10 c=6
a=16 b=11 d=26
a/b=1
a%b=5
a*= b =176
0
1
```

**প্রয়োগ ২.** Find out the output of the following program: [North-West Power generation Company Ltd. - AE -2018]

```
#include<iostream>
using namespace std;
int main(){
    int i=1, j=1, k=1;
    cout<<++i||++j&&++k;
    cout<<" "<<i<<" "<<j<<" "<<k;
    return 0;
}
```

**Output:** 2 2 1 1

**প্রয়োগ ৩.** Find out the output of the following program: [BD Telecommunication Company Ltd. - AM -2018]

```
#include<stdio.h>
#define n 7
int main(){
    char str[n] = "abpqx\n";
    int i=0;
    for(i=0; i<n-2; i++){
        if(i%2==0){
            printf("%c", str[i++]);
        }
        else
            printf("%c", str[i--]);
    }
    return 0;
}
```

**Output:** abpqx

**প্রয়োগ ৪.** Find out the output of the following program: [BD Computer Council - AP -2018]

```
#include<iostream>
using namespace std;
int main(){
    int i=-20, j=-1, k=2, m;
    m=++i&&++j&&++k;
    cout<<i<<" "<<j<<" "<<k<<" "<<m;
    return 0;
}
```

**Output:** -19 0 2 0

**প্রয়োগ ৫.** Find out the output of the following program: [Palli Sanchay Bank- AP -2018]

```
#include<stdio.h>
#include<conio.h>
int main(){
    int n=10;
    for(;;)
        printf("%d", n);
    return 0;
}
```

**Output:** Infinite loop

**প্রয়োগ ৬.** Find out the output of the following program: [Palli Sanchay Bank- AP -2018]

```
#include<stdio.h>
#include<conio.h>
int main(){
    int i=2, j=2;
    while(i-->i:j++)
        {
            printf("%d", i);
        }
    return 0;
}
```

**Output:** 1

**প্রয়োগ ৭.** Find out the output of the following program: [Palli Sanchay Bank- ADA -2018]

```
#include<stdio.h>
int main(){
    int a=2, b=2, c=1;
    if(a&b>c)
        printf("Bangladesh");
    else
        printf("not Bangladesh");
    return 0;
}
```

**Output:** Bangladesh

**প্রয়োগ ৮.** Find out the output of the following program: [Palli Sanchay Bank- Programmer -2018]

```
#include<stdio.h>
#include<conio.h>
int main(){
    char str[120] = "Digital Bangladesh";
    int n;
```

```
n=strlen(str);
str[4]='\0';
printf("%s", str);
return 0;
}
```

**Output:** Digi

**প্রয়োগ ৯.** Find out the output of the following program: [Palli Sanchay Bank- Programmer -2018, Jibon Bima Corporation- AP -2018]

```
#include<stdio.h>
#include<conio.h>
int main(){
    int i;
    for(i=0; i<5; i++){
        if(i==3){
            continue;
        }
        printf("%d", i);
    }
    return 0;
}
```

**Output:** 0 1 2 4

**প্রয়োগ ১০.** Find out the output of the following program: [BD Computer Council - AP -2018]

```
#include<stdio.h>
int main(){
    int a=0, b=1, c=2;
    *((a)?&b:&a)=a?b:c;
    printf("%d %d %d", a,b,c);
    return 0;
}
```

**Output:** 2 1 2

**প্রয়োগ ১১.** Find out the output of the following program: [BD Computer Council - AP -2018]

```
#include<stdio.h>
int main(){
    float n=2;
    switch(n) {
        case 2:
            printf("Hiæ");
            break;
        default:
            printf("Hello");
    }
    return 0;
}
```

**Output:** error: switch quantity not an integer

**প্রয়োগ ১২.** Find out the output of the following program: [DESCO -assistant programmer- 2016]

```
#include<stdio.h>
#include<conio.h>
#define max(a,b) (a>b?a:b)
int main(){
    int i=1, j=2, k=0;
    k=max(++i, ++j);
    printf("%d", k);
    return 0;
}
```

**Output:** 4

**প্রয়োগ ১৩.** নিচের প্রয়োগের Output কি হবে ?

```
#include<iostream>
using namespace std;
int main(){
    long int n;
    int i,j;
    n=1;
    for(i=1;i<=2;i++)
        for(j=1;j<=5;j++)
            n=n*i+j;
    cout<<n;
    return 0;
}
```

**Output:** 569

**প্রয়োগ ১৪.** নিচের programming টির Output লেখ :

```
main ()
{
    float a,b,c,x,y,z;
    a=9;
    b=12;
    c=3;
    x=a-b/3+c*2-1;
    y=a-b/(3+c)*(2-1);
    z=a-(b/(3+c)*2)-1;
    printf("X=%f\n",x);
    printf("Y=%f\n",y);
    printf("Z=%f\n",z);
    return 0;
}
```

**Output:**  
X=10.000000  
Y=7.000000  
Z=4.000000

প্রয়োগ ১৫. What is the output of the following code? [DESCO - AE -2019, NPCBL - Junior Technical Engineer - 2019]

```
#include<stdio.h>
#define x 9+2/4*3-2*4+(5-4)*3

int main(){
    int i, y;
    y=6+3*3/5;
    i=x*x+y;
    printf("%d", i);
    return 0;
}
```

**Output:** 30

Here,  $y = 6+3*3/5 = 6+1 = 7$   
 $i=9+2/4*3-2*4+(5-4)*3 = 9+0-8+27 = 28$   
 $2*4+(5-4)*3+7 = 9+0-8+27 = 28$

প্রয়োগ ১৬. Find the mistake in the following program and write the correct form. [Combined Five Bank - AME - 2019, DBBL - Probationary Officer - 2017]

- a. `unsigned I;`
- b. `for(i=100; i<=0; -i)`
- c. `printf("%d", i);`

Correct Form is following-

- a. `int i;`
- b. `for(i=100; i>=0; i--)`
- c. `printf("%d", i);`

প্রয়োগ ১৭. Rewrite the following C program using do-while loop. [JU-2014]

```
#include<stdio.h>
Main(){
    Int count;
    Count=0;
    While(count<100){
        Count++;
        Printf("Hello World\n");
    }
}
```

**Ans:**

```
#include <stdio.h>
int main(){
    int Count=0;
    do{
        Count++;
        printf("Hello World\n");
    }while(Count<100);
}
```

প্রয়োগ 18. What you will see if you run the following codes. [JU-2014]

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

```
int main(void){
    int first=15, second=10, sum;
    printf("Enter two Integer:");
    scanf("%d %d", &first, &second);
    printf("The two Numbers are: %d\n", first, second);
    sum=first+second;
    printf("Their Sum is: %d", sum);
}
Output:
Enter two Integer: 12 10
The two Numbers are: 12 10
Their Sum is: 22
```

প্রয়োগ ১৯. Find the output of the following program. [Janata Bank-2016]

Program:

```
#include <stdio.h>
int main(){
    int i=0;
    int x=i++, y=++i;
    printf("%d %d", x,y);
return 0;
}
```

**Output:** 0 2

প্রয়োগ ২০. What is the output of following program? [Investment Corporation of BD - AP -2019]

```
#include<stdio.h>
int main(){
    printf("%d\t", sizeof(6.5));
    printf("%d\t", sizeof(9000));
    printf("%d\t", sizeof("A"));
    return 0;
}
```

**Output:** 8 4 2

প্রয়োগ ২১. How many number will be printed?

[Investment Corporation of BD - AP -2019]

```
#include<stdio.h>
int main(){
    int i=5;
    while(i>=0){
        printf("%d ", i);
        i=i-1;
    }
    return 0;
}
```

**Output:** 5 4 3 2 1 0

প্রয়োগ ২২. What is the output of the following code? [BD Completion Commission- AP -2019]

```
#include<stdio.h>
#include<math.h>
int main(){
    char f[]={ "Ronaldo", "Messi",
    "Zidan", "Maradona"}, str[20];
    printf("%s\n", f[1]+2);
    printf("%s", f[2]+1);
    return 0;
}
```

Output:  
ssi  
idan

প্রয়োগ ২৩. What is the output of the following code?

[DBBL - Assistant Network/Hardware Engineer -2019]

```
int a=3; int b=10;
if(a>b)
    printf("A");
else
    printf("B");
```

**Output:** B

প্রয়োগ ২৪. What is the output of the following code?

[DBBL - Assistant Network/Hardware Engineer -2019]

```
int i=0;
int n=0;
while(i!=n){
    if(i==3)
        continue;
    printf("%d", i);
    i++;
}
```

**Output:** No values will return here. Because, the first condition of while loop is false.

প্রয়োগ ২৫. Correct the following code. [Bangabandhu Sheikh Mujibur Rahman Aviation and Aerospace University - AP&ASE -2019]

```
#include <stdio.h>
int main(){
    int pass, x=10;
    if(pass=1234) {
        print("Correct password");
        x=0;
        printf("Wrong password, try
another");
    }
    return 0;
}
```

**Solution:**

```
#include <stdio.h>
int main()
{
    int pass, x=10;
    printf("Enter a password: ");
    scanf("%d", &pass);
    if(pass==1234){
        printf("Correct password");
        x=0;
    }
    else
        printf("Wrong password, try
another");
    return 0;
}
```

প্রয়োগ ২৬. What will be the output of the program?

[BREB - Assistant Software Engineer -2019]

```
#include<stdio.h>
int main(){
    unsigned int res;
    res = (64>>(2+1-2))&(~(1<<2));
    printf("%d\n", res);
    return 0;
}
```

**Output:** 32

প্রয়োগ ২৭. What will be the output of the program if the array begins 1200 in memory? [BREB - Assistant Software Engineer -2019]

```
#include <stdio.h>
int main(){
    int arr[]={2,3,4,1,6};
    printf("%u, %u,
%u\n", arr,&arr[0],&arr);
    return 0;
}
```

**Output:** 1200, 1200, 1200

প্রয়োগ ২৮. What will be the output of the program?

[BREB - Assistant Software Engineer -2019]

```
#include <stdio.h>
int main(){
    union ver{
        int a,b;
    };
    union ver v;
    v.a=10;
    v.b=20;
    printf("%d\n", v.a);
    return 0;
}
```

**Output:** 20

### Structure Programming (C Program)

প্রোবলেম ২৯. If the size of pointer is 4 bytes then what will be the output of the program? [BREB - Assistant Software Engineer -2019]

```
#include<stdio.h>
int main(){
    char *str[]={ "Frogs", "Do", "Not", "Die",
    "They", "Croak! "};
    printf("%d %d", sizeof(str),
    strlen(str[0]));
    return 0;
}
```

**Output:** 48, 5 [Note: result depend on compiler]

প্রোবলেম ৩০. What is the write code of the following statement in while loop format.

```
for(a=1; a&1; t=100; a++)
    Printf("%d\n", a*a);
[Sonali/Janata Bank - Officer (ICT) - 2019]
```

Ans:

```
int a=1;
while(a<=100){
    printf("%d\n", a*a);
    a++;
}
```

প্রোবলেম ৩১. Find the output of the following program. Program: [DU-2017, BD Krishi Bank - AP - 2018]

```
#include <stdio.h>
int main(){
    int i,j;
    for(i=0;i<10;i++){
        for(j=10;j>0;j--){
            printf("*");
            if(j==i+1)
                break;
        }
    }
    return 0;
}
```

**Output:**\*\*\*\*\*

প্রোবলেম ৩২. Find out the output of the following program: [DESCO -assistant programmer- 2016]

```
include<stdio.h>
#include<conio.h>
int sum(int i){
    static int total=0;
    total+=i;
    return total;
}
int main(){
    int i;
```

```
    for(i=1; i<10; i++) {
        printf("%d ", sum(i));
    }
    return 0;
}
```

**Output:** 13 6 10 15 21 28 36 45

প্রোবলেম ৩৩. Find out the output of the following program: [Jibon Bima Corporation- AP -2018]

```
#include<stdio.h>
#include<conio.h>
int main(){
    char str1[120] = "Bangladesh";
    char str2;
    str1[4] = '\0';
    str2 = strlen(str1);
    printf("%s", str1);
    return 0;
}
```

**Output:** Bang

প্রোবলেম ৩৪. Find the output of the following program.

Program: [BWDB-2017]

```
#include <stdio.h>
int main(){
    int i=1,j=1,k=0,l=2,m;
    m=i++&&j++&&k++|| i++;
    printf("%d%d%d%d", i, j, k, l, m);
    return 0;
}
```

**Output:** 32121

প্রোবলেম ৩৫. Consider the following program : [BTEB- 2005]

```
Main (){
    int i,j;
    for (i=1; i < =2; i++){
        for (j = 1; j < =2; j++)
            printf ("\n%d\n", i, j);
    }
}
```

Now modify the above program so that it will print only 1 2 and 2 1.

Ans:

```
Main (){
    int i,j;
    for (i=1; i < =2; i++){
        for (j = 1; j < =2; j++)
            if(i==j)
                continue;
            printf("\n%d %d\n", i, j);
    }
}
```

### Structure Programming (C Program)

**Output:** 1 2 and 2 1.

প্রোবলেম ৩৬. Find the output of the C Program. [BCC- AP-2021]

```
#include<stdio.h>
int func(int x[], int i){
    int s = x[i];
    if(i>0){
        s += func(x, i-1);
    }
    printf("%d ", s);
    return s;
}
```

```
int main(){
    int arr[4]={1,3,2,4};
    func(arr,2);
    return 0;
}
```

প্রোবলেম ৩৭. 5 input are 165,85,528,409,1200. DPDC wants to classify users on their consumed electricity units. Write a c/c++ program that will take int n input. Then program will take as int input n number of electricity unit consumed in reverse order of the input. The program will print to the class of. [BCC-AP-2021, DPDC JAM(ICT)-2020]

Input	Output
Consumed>1000 Heavy Industrial consumer.	1200 = Heavy Industrial consumer.
500<=consumed <1000 Industrial consumer.	409 = commercial consumer.
100<=consumed <500 commercial consumer.	528 = Industrial consumer.
Consumed <100 Domestic consumer.	85 = Domestic consumer.
	165 = commercial consumer.

Ans:

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main(){
    int n,j=0;
    int arr[100];
    printf(" Enter the number of Consumers:");
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        printf(" Enter %dst consumer consumed Unit:",(i+1));
        scanf("%d",&arr[i]);
    }
    printf(" Output:\n");
    for(int i=n-1;i>=0;i--){
        if(arr[i]>1000)
            printf(" %d is Heavy Industrial consumer.\n",arr[i]);
        else if(arr[i]>=500 & arr[i]<1000)
            printf(" %d is Industrial consumer.\n",arr[i]);
        else if(arr[i]>=100 && arr[i]<500)
            printf(" %d is commercial consumer.\n",arr[i]);
        else if(arr[i]<100)
            printf(" %d is Domestic consumer.\n",arr[i]);
        else
    }
}
```

### Structure Programming (C Program)

```

        printf("%d is Special type consumer.\n", arr[i]);
    }
    printf("\n");
}
return 0;
}

Output:
Enter the number of Consumers:5
Enter 1st consumer consumed Unit:165
Enter 2st consumer consumed Unit:85
Enter 3st consumer consumed Unit:528
Enter 4st consumer consumed Unit:409
Enter 5st consumer consumed Unit:1200
Output:
1200 is Heavy Industrial consumer.
409 is commercial consumer.
528 is Industrial consumer.
85 is Domestic consumer.
165 is commercial consumer.

প্রোগ্রাম ৭০. Binary search program in C [বরিশাল ইউনিয়ন কলেজ- ২০২১]
উত্তর: নিচে বাইনারি সার্চের প্রোগ্রাম দেওয়া হলো:
#include <stdio.h>
int main(){
    int c, first, last, middle, n, search, array[100];

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);
    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    printf("Enter value to find\n");
    scanf("%d", &search);
    first = 0;
    last = n - 1;
    middle = (first+last)/2;

    while (first <= last) {
        if (array[middle] < search)
            first = middle + 1;
        else if (array[middle] == search) {
            printf("%d found at location %d.\n", search, middle+1);
            break;
        }
        else
            last = middle - 1;
        middle = (first + last)/2;
    }

    if (first > last)
        printf("Not found! %d isn't present in the list.\n", search);
    return 0;
}

```

```

Enter number of elements
7
Enter 7 integers
1
4
5
8
9
11
43
Enter value to find
11
11 found at location 5.

```

### Structure Programming (C Program)

**প্রোগ্রাম ৮০. write a pseudo code to generate all possible permutations for a given word. [বরিশাল ইউনিয়ন কলেজ- ২০২১]**

উত্তর: নিম্নে একটি শব্দের সকল পারমুটেশন নির্ণয়ের প্রোগ্রাম দেওয়া হলো:

```

#include <bits/stdc++.h>
using namespace std;
void permute(string a, int l, int r) {
    if (l == r)
        cout << a << endl;
    else {
        for (int i = l; i <= r; i++) {
            swap(a[l], a[i]);
            permute(a, l+1, r);
            swap(a[l], a[i]);
        }
    }
}

int main() {
    string str = "ABC";
    int n = str.size();
    permute(str, 0, n-1);
    return 0;
}

```

**OUTPUT:**

ABC  
ACB  
BAC  
BCA  
CBA  
CAB

**প্রোগ্রাম ৮১. নিচের C প্রোগ্রামটির ভূলগুলো সঠিক করুন এবং প্রোগ্রামটির আউটপুট লিখুন। [বানিজ্য মহানালয়- ২০২১-সহকারী প্রোগ্রাম (৮ন-কাডার)]**

উত্তর:

```

#include<stdio.h>
int main{
    int I,sum=0
    for(i=1,i<=10,i+
    +){
        sum=sum+i
    }
    printf(sum of
    number=d,sum)
    return 0
}

```

আউটপুট: sum of number: 55

**প্রোগ্রাম ৮২. Given a IP4 address String, write C/C++/JAVA code to show the class the IP address belongs to. [RUBALI BANK ANE-2021]**

Sample Input: 192.168.0.0  
Sample Output: Class C

```

#include<bits/stdc++.h>
using namespace std;
int main(){
    char input[100];
    gets(input);
    int sum=0,i=0;
    while(input[i]!='.'){
        sum=sum*10+input[i]- '0';
        //অথবা sum=sum*10+input[i]-48;
        i++;
    }
    if(sum<=127)cout<<"A";
    else if(sum<=191)cout<<"B";
    else if(sum<=223)cout<<"C";
    else if(sum<=239)cout<<"D";
    else if(sum<=255)cout<<"E";
    else cout<<"Wrong";
}

```

input[i] contains a character that represents a digit as for example '2'. To convert this character that for example in ASCII has internal code equal to 50 (while internal code of character '0' is 48) there is used expression:  
input[i] - '0' that is equivalent to  
50 - 48 and equal to number 2

প্রয়োগ ৮০. You have a file name accounts.txt which contain the following information. Now write a C/C++/Java program to find the following.

Ac	Name	Type	Balance
101	Tom	Saving	12000
102	Jerry	Current	5000
103	Kot	Saving	10000
104	Tom	Current	5000
105	Kot	Saving	20000

- Total balance of saving account
- Find the highest and second highest balance of saving account

Solution:

```
#include<stdlib.h>
#include<iostream>
#include<stdio.h>
#include<string.h>
using namespace std;

int main(){
    int acno,balance,savings_balance=0;
    char name[50], types[30];
    int savings_highest=0,savings_second_higest=0;

    freopen("accounts.txt", "r",stdin);
    while(scanf("%d %s %d",&acno,&name,&types,&balance)==4){
        if(strcmp(types,"Savings")==0){
            savings_balance+=balance;
            if(balance > savings_highest){
                savings_second_higest=savings_highest;
                savings_highest=balance;
            }
            if(balance<savings_highest&&balance>savings_second_higest){
                savings_second_higest=balance;
            }
        }
        printf("Balance of Saving accounts=%d\n",savings_balance);
        printf("Highest Balance of Savings Accounts=%d\n",savings_highest);
        printf("Second Highest Balance of Savings Accounts=%d\n",savings_second_higest);
    }
    return 0;
}
```

প্রয়োগ ৮১. Find the output of the following program.

```
#include <stdio.h>
int main(){
    int i=0,j=5,x=0,count=0;
    while(j>i) {
        if(i==7)
            break;
        x=x+i+count;
        count=count+2;
    }
}
```

```
i++;
}
printf("i= %d, count=%d",i,count);
printf("j= %d, x=%d",i,x);
return 0;
}
```

Solution:

Steps	j	i	count	x	Increment
1	initial j=5	i=0	initial count=0	initial =0	
2	j>i true	i==7 false	count=count+2	x=x+i+count	i++
3	j>i true	i==7 false	count=4	x=0+0+0	i=1
4	j>i true	i==7 false	count=6	x=3	i=2
5	j>i true	i==7 false	count=8	x=9	i=3
6	j>i true	i==7 false	count=10	x=18	i=4
				x=30	i=5
<b>Output</b> i= 5, count=10, j= 5, x=30					

প্রয়োগ ৮২. Find out the output of the following program. [SGFL-2021-AE]

```
#include<stdlib.h>
#include<string.h>
int main(){
    int i=0, length;
    char string[]="Hello\0, World!!";
    length = strlen(string);
    char *s = string;
    for(i=0; i<length; ++i)
        printf("%c", *++s);
    return 0;
}
```

Output:  
G:\Practice Program\output\_program1.exe  
Hello  
Process returned 0 (0x0) execution time : 0.189 s  
Press any key to continue.

প্রয়োগ ৮৩. How many times the loop given below will get executed? [বিভিন্ন ম্যানালয়-২০১৭]

```
void main(){
    int i;
    for(i=20, i=10; i<=20; i++)
        printf("in %d",i)
}
```

ANS: 11 times

প্রয়োগ ৮৪. (a) Suppose, you have purchased 20 different items from a super shop. Write a program to finds the item with the minimum price (Consider that item prices are unique). [তুমি সুপার সোপ থেকে ২০টা আইটেম কিনেছো, এর মধ্যে সবচেয়ে কম দামের আইটেমটি বের করার প্রয়োগ লিখ,- প্রতিটি আইটেমের দাম ইউনিক হবে মানে আলাদা আলাদা হবে।]

(b) Explain the time-complexity of an algorithm. [বিভিন্ন ম্যানালয়-২০১৭]

উত্তর:  
#include<stdio.h>
int main(){
 int a[50], i, small;//1
 for(i=0; i<20; i++)//20
 scanf("%d", &a[i]);
 small=a[0];//1
 for(i=1; i<20; i++)//20
 if(small>a[i])
 small=a[i];
 printf("Smallest element: %d", small);
 return 0;
}

Time-complexity of an algorithm O(1)+O(20)+  
O(1)+O(20)=O(20)

প্রশ্ন ৪৮. Write Output from below code:

```
#include <stdio.h>
int main(){
    char s[] = "Bangladesh Industrial and
    Technical Assistant Center";
    char *s1;
    s1=s;
    for(int i=0; i<10; i++){
        printf(" %c ", s[i]);
        ++s1;
    }
    printf("\n");

    for(int i=0; i<10; i++){
        printf(" %c ", s1[i]);
        ++s1;
    }
    return 0;
}
```

Output:

```
B a n g l a d e s h
n u t i l a d T c
```

#### Object Oriented Programming (C++/Java)

[**Syllabus:** BPSC CS: Introductory programming with C++/JAVA. Introduction to object oriented programming: Encapsulation, inheritance and polymorphism, Mechanic Language Programming, Template functions and classes multi-threads exceptions, Class and object.

**NTRCA ICT:** Topics include object-oriented programming concepts, such as classes, objects, methods, interfaces, packages, inheritance, encapsulation, and polymorphism.]