অধ্যায়-৩ অ্যারে, পয়েন্টার এবং স্ট্রিং

SOS অতি সংক্ষিপ্ত প্রশ্নোত্তর ঃ

***** ১। লিনিয়ার অ্যারে (Linear Array) বলতে কী বুঝায়?**

বাকাশিবো- ২০০২, ০৬

উত্তর ঃ একই টাইপের একাধিক ভেরিয়েবলের সমষ্টিকে অ্যারে বলে। যেখানে ডাটাগুলো পরপর অবস্থান করে। ইহা Linear Data Structure, Array এর Indexing সবসময় 0 থেকে শুরু হয়।

Array ঘোষণার Syntax:

```
Data_Type Array_Name [Size];
i. e : int a [10];
```

**** ২। One Dimensional Array কাকে বলে?**

উত্তর ঃ যে সকল অ্যারের একটিমাত্র Index বা Subscript থাকে, তাকে One Dimensional Array বা একমাত্রিক অ্যারে বলে।

Syntax:

```
Date_Type Array_Name [Size];
i. e : int a [10];
```

**** ৩। Two-Dimensional Array কী?**

উত্তর ঃ যে সকল অ্যারের দুইটি Subscript থাকে, তাকে Two Dimensional (2D) বা দ্বিমাত্রিক অ্যারে বলে।

Syntax:

Date_Type Array_Name [row] [column];

i. e: int a [2] [3];

**** ৪। Array এর Length বের করার সূত্রটি লেখ।** বাকাশিবো- ২০১২

উত্তর ঃ Array এর Length বা উপাদান সংখ্যা = UB - LB + 1

যেখানে, UB = Upper Bound (সর্বশেষ উপাদান)

LB = Lower Bound (প্রথম উপাদান)

**** ৫। Location নির্ণয়ের সূত্রটি লেখ, যা Linear Array এর K-তম উপাদান।** বাকাশিবো- ২০২৩

উত্তর ঃ সূত্র ঃ Loc (LA[K]) = Base (LA) + W(K–LB)

এখানে, LOC = Location

LA = Linear Array

K = যে উপাদানের Address বের করতে হবে। Base = Base Address বা Starting Address

W = Word Size (প্রতি উপাদানের জন্য মেমরিতে বরাদ্দকৃত জায়গা)

LB = Lower Bound

***** ৬। পয়েন্টার (Pointer) কাকে বলে?** বাকাশিবো- ২০০৭, ১৪'পরি

উত্তর ঃ পয়েন্টার হলো এক বিশেষ ধরনের ভেরিয়েবল, যা নির্দিষ্ট টাইপের ভেরিয়েবলের অ্যাড্রেস ধারণ করে। পয়েন্টার ব্যবহার করে একজন প্রোগ্রামার সরাসরি Memory Address নিয়ে কাজ করতে পারেন।

Syntax:

Date_Type * Pointer_Name;

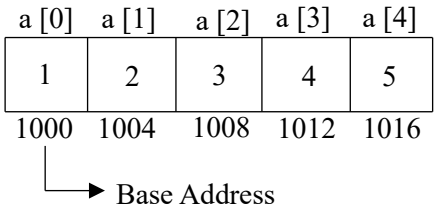
i. e : int * a;

***** ৭। Pointer Array কী?** বাকাশিবো- ২০০৫, ১৩' পরি, ১৭, ২১

উত্তর ঃ Pointer কে যদি কোনো অ্যারের মাধ্যমে সজ্জিত করা হয়, তবে তাকে Pointer Array বলে।

যেমন ঃ int a [5] = {1, 2, 3, 4, 5};

int&p;



ধরে নিলাম, Base Address = 1000

int এর জন্য Memory Size = 4 byte

**** ৮। Jagged Array বলতে কী বুঝায়?** বাকাশিবো- ২০১২, ১৫'পরি, ১৯

উত্তর ঃ Jagged Array এক বিশেষ ধরনের Two Dimensional Array, যেখানে Row এর সাইজ নির্দিষ্ট থাকবে কিন্তু প্রতিটি Row এর Column এর সাইজ প্রয়োজন অনুযায়ী কম বেশি করা যাবে। ইহাকে Ragged Array ও বলা হয়।

যেমন ঃ

```
int [ ] [ ] a = new int [2] [ ];
a [0] = new int [3];
a [1] = new int [2]
```

00	01	02
10	11	

এখানে, ১ম Row তে 3টি কলাম এবং ২য় Row তে দুইটি কলাম।

***** ৯। Field, Record ও File কাকে বলে?**

BPSC- 2014, বাকাশিবো- ২০০২, ০৪, ০৬, ০৮, ০৯, ১০, ১০'পরি, ১২, ১২'পরি, ১৩

উত্তর ঃ Field: একই জাতীয় ডাটা উপাদানকে Field বলে।

যেমন ঃ নাম, রোল, সেমিস্টার ইত্যাদি।

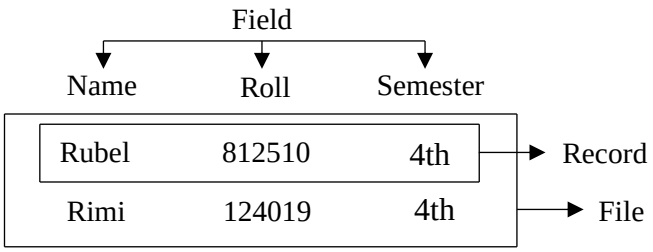
Record: একাধিক Field এর সমষ্টিকে Record বলে।

যেমন ঃ একজন পরিক্ষার্থীর নাম, রোল এবং সেমিস্টার এই তিন Field এর সমন্বয়ে একটি Record গঠিত।

File: পরস্পর সম্পর্কযুক্ত একাধিক Record এর সমষ্টিকে File বলে।

যেমন ঃ তোমার ক্লাসের সকল শিক্ষার্থীর নাম, রোল এবং সেমিস্টারের সকল Record নিয়ে একটি File গঠিত হয়।

উদাহরণ ঃ



***** ১০। String কাকে বলে?**

উত্তর ঃ এক বা একাধিক Character এর সমষ্টিকে String বলে। যা double quotation (”) এর মাঝে লিখতে হয়।

যেমন ঃ char str1 [ ] = “Softmax online school” ;

সহজ ভাষায় String হচ্ছে Character Type এর Array.

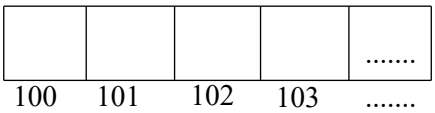
SOS সংক্ষিপ্ত প্রশ্নোত্তর ঃ

***** ১। Linear Array কে Memory তে উপস্থাপন করার কৌশল বর্ণনা কর।** বাকাশিবো- ২০০২, ০৭, ০৯, ১১, ১২, ১২’পরি, ১৩, ১৩’পরি, ১৬’পরি

উত্তর ঃ মনে করি, LA একটি Linear Array, যা কম্পিউটার মেমরিতে আছে।

আমরা জানি, কম্পিউটার মেমরিতে সংরক্ষিত অ্যারের উপাদানসমূহকে খুব সহজেই মেমরির পর্যায়ক্রমিক Address দ্বারা উপস্থাপন করা যায়। Array এর শুরুর Address অথবা Base Address জানা থাকলে ক্রমিক Address দ্বারা সকল উপাদানসমূহের Address বের করা যায়। Array এর যেকোনো Location এর Address নিচের সূত্র দ্বারা বের করা যায়।

$$\mathbf{LOC\ (LA[K]) = Base\ (LA) + W(K-LB)}$$



এখানে, LOC = Location

LA = Linear Array

K = যে উপাদানের Address বের করতে হবে।

Base = Base Address বা Starting Address

W = Word Size (প্রতি উপাদানের জন্য মেমরিতে বরাদ্দকৃত জায়গা)

LB = Lower Bound

Linear Array এর প্রতিটি উপাদানের Address কম্পিউটারের খুঁজে বের করার প্রয়োজন হয় না। Base Address জানা থাকলেই যে কোনো Location খুঁজে বের করা যায়।

একটি Array তে উপাদান সংখ্যা বের করার জন্য নিচের সূত্র ব্যবহার করা হয় ঃ

Array এর উপাদান সংখ্যা = UB - LB + 1

***** ২। Linear Array Traversing এর Algorithm লেখ।**

বাকাশিবো- ২০০০, ০৫, ০৭, ০৯, ১১, ১২, ১৩' পরি, ১৭' পরি, ১৮' পরি, ১৯, ২১, ২২

উত্তর ঃ Algorithm:

Step 1: Start

Step 2: Initialize counter set K = LB

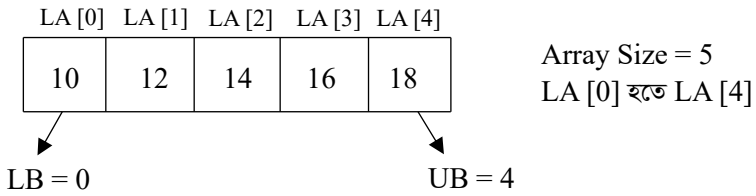
Step 3: Repeat steps 4 & 5 while K $\leq$ UB

Step 4: Apply process to LA [K]

Step 5: Increment counter K = K + 1

Step 6: Stop

ব্যাখ্যা ঃ

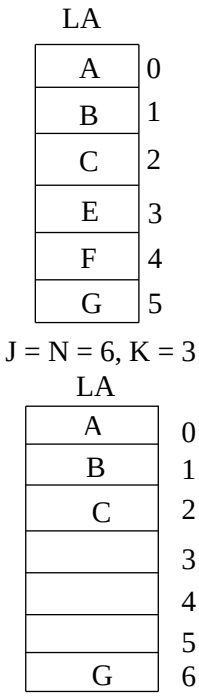


প্রথমে Counter এর প্রারম্ভিক মান K = LB = 0 নির্ধারণ করে দেওয়া আছে। এরপর While loop ব্যবহার করে যতক্ষণ পর্যন্ত K এর মান UB এর চেয়ে ছোট বা সমান থাকবে ততক্ষণ পর্যন্ত পরবর্তী দুই ধাপ অর্থাৎ LA [K] উপাদানের মান Access বা Traversing করতে এবং পরবর্তী Location এ যাওয়ার জন্য K এর মানের সাথে 1 যোগ করবে।

***** ৩। Linear Array তে Data Insert বা সংযোজনের Algorithm লেখ।** বাকাশিবো- ২০০৬, ০৭, ০৯, ০৯'পরি, ১২, ১৬, ১৭'পরি, ১৮'পরি প্রবাসী কল্যাণ ও বৈদেশিক কর্মসংস্থান জনশক্তি কর্মসংস্থান ব্যুরো- ২০১৮

উত্তর ঃ Algorithm:

Step 1: Start
Step 2: Initialize counter set J = N
Step 3: Repeat Steps 4 & 5 while J $\geq$ K
Step 4: Move Jth element downward
    Set LA [J + 1] = LA [J]
Step 5: Decrease the counter, Set J = J-1
Step 6: Insert element LA [K] = ITEM
Step 7: Update N = N+1
Step 8: Stop



ব্যাখ্যা ঃ এখানে N = Array size
J = Counter variable
K = যে location এ নতুন Data insert করব
ITEM = নতুন Data

Counter এর মান N এ Initialize করতে হবে। একটি মানকে এক location নিচে বসিয়ে

Counter এর মান 1 বাড়াতে হবে। এভাবে 4 ও 5 নং Step চলবে। এরপর K তম অবস্থান

ITEM Insert করব। নতুন Data insert হওয়ায় Size 1 বেড়ে গেল। তাই N = N + 1 করা হয়েছে।

***** ৪। Linear Array হতে Data Delete করার Algorithm লেখ।** **বাকাশিবো- ২০০৩, ০৬'পরি, ১২, ১৪'পরি, ১৯**

উত্তর ঃ **Algorithm:**

Step 1: Start
Step 2: Set ITEM = LA [K]
Step 3: Repeat for J = K to N-1
    Move (J+1) element upward
    Set LA [J] = LA [J+1]
    (End of loop)
Step 4: Update N = N-1
Step 5: Stop

LA	
A	0
B	1
C	2
D	3
E	4

ব্যাখ্যা ঃ প্রথম কাজ হচ্ছে, যে উপাদানটি (ITEM) Delete করব, তাকে আগে খুঁজে বের করতে হবে। ধরি, 2 location এর মান Linear Array (LA) হতে একবার করে উপরে উঠবে। এভাবে পার্যায়ক্রমে Looping এর শর্ত পর্যন্ত চলবে। একটি Data delete হবার পর মোট Data হতে একটি Data কমাতে $n = n - 1$ করা হয়েছে।

***** ৫। Matrix Multiplication (ম্যাট্রিক্স গুণন) এর Algorithm লেখ।** বাকাশিবো- ২০০৬, ১০'পরি, ১২, ১৩'পরি, ১৪, ১৪'পরি, ১৬'পরি, ১৯, ২৩

উত্তর ঃ Algorithm:

Step 1: Repeat steps 2 and 3 for I = 1 to M
Step 2: Repeat steps 3 and 4 for J = 1 to N
Step 3: Set C [I, J] = 0
Step 4: Repeat for K = 1 to P

$$C[I,J] = C[I,J] + A[I,K] * B[K,J]$$

Step 5: Exit

ব্যাখ্যা ঃ Matrix Multiplication (গুণনের) শর্ত ঃ

প্রথম Matrix এর Column সংখ্যা, দ্বিতীয় Matrix এর Row এর সংখ্যা সমান হলে Matrix Multiplication করা যাবে।

উদাহরণ ঃ $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}_{2\times2}$,

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix}_{2\times3}$$

A=M×P
[M=I, P=K]
B=P×N
[P=K, N=J]
C=M×N [I,J]

$$C = AB$$

C এর Order বা মাত্রা হবে $= 2 \times 3$

$$C_{11} = (A_{11} \times B_{11}) + (A_{12} \times B_{21})$$
$$C_{12} = (A_{11} \times B_{12}) + (A_{12} \times B_{22})$$
$$C_{13} = (A_{11} \times B_{13}) + (A_{12} \times B_{23})$$

$$C_{21} = (A_{21} \times B_{11}) + (A_{22} \times B_{21})$$
$$C_{22} = (A_{21} \times B_{12}) + (A_{22} \times B_{22})$$
$$C_{23} = (A_{21} \times B_{13}) + (A_{22} \times B_{23})$$
$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \end{bmatrix}$$

SOS রচনামূলক প্রশ্নোত্তর ঃ

**** ১। দ্বিমাত্রিক অ্যারেকে মেমরিতে উপস্থাপন পদ্ধতি বর্ণনা কর।**

বাকাশিবো- ২০০৭, ২৩

উত্তর ঃ দ্বিমাত্রিক অ্যারে ঃ যে সকল অ্যারের দুইটি Subscript থাকে, তাকে Two Dimensional (2D) বা দ্বিমাত্রিক অ্যারে বলে।

দ্বিমাত্রিক অ্যারেকে মেমরিতে উপস্থাপন ঃ দ্বিমাত্রিক অ্যারেকে কম্পিউটার মেমরিতে দুই রকমভাবে উপস্থাপন করা যায়। যথা ঃ

i) Row Major Order (রো ভিত্তিক) – Row × Row

ii) Column Major Order (সারি ভিত্তিক) – Column × Column

i) Row Major Order ঃ

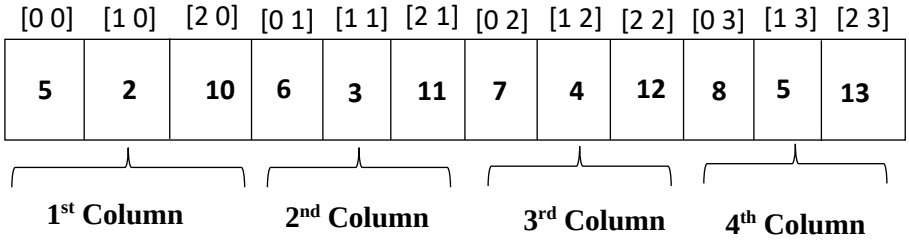
[0 0]	[0 1]	[0 2]	[0 3]	[1 0]	[1 1]	[1 2]	[1 3]	[2 0]	[2 1]	[2 2]	[2 3]
5	6	7	8	2	3	4	5	10	11	12	13
1st Row				2nd Row				3rd Row			

মনে করি, 4 × 3 Size এর একটি দ্বিমাত্রিক অ্যারেতে নিম্নোক্ত Data গুলি আছে।

int a [3] [4];

	0	1	2	3
0	5	6	7	8
1	2	3	4	5
2	10	11	12	13

ii) Column Major Order ঃ



এখানে, M = Row size

N = Column size

BA = Base Address

J = যে Row এর মান বের করতে হবে।

K = যে Column এর মান বের করতে হবে।

W = Word size

নিচের দুইটি সূত্র ব্যবহার করে Row Major ও Column Major অনুসারে যেকোনো উপাদান বের করা যায়।

Row Major Order ঃ

LOC (A [J,K]) = Base (A) + W [N (J-1) + (K-1)]

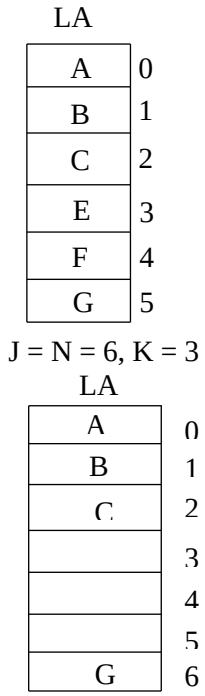
Column Major Order ঃ

LOC (A [J,K]) = Base (A) + W [M (K-1) + (J-1)]

***** ২। Linear Array তে Data Insert বা সংযোজন এবং Data Delete বা বিয়োজনের Algorithm লেখ।**

বাকাশিবো- ২০১২, ১৩'পরি, ১৭, ১৭'পরি, ১৮

উত্তর ঃ Linear Array তে Data Insert বা সংযোজনের Algorithm নিম্নরূপ ঃ



Algorithm:

Step 1: Start
Step 2: Initialize counter set J = N
Step 3: Repeat Steps 4 & 5 while J $\geq$ K
Step 4: Move Jth element downward
Set LA [J + 1] = LA [J]
Step 5: Decrease the counter, Set J = J-1
Step 6: Insert element LA [K] = ITEM
Step 7: Update N = N+1
Step 8: Stop

ব্যাখ্যা ঃ এখানে N = Array size
J = Counter variable
K = যে location এ নতুন Data insert করব
ITEM = নতুন Data

Counter এর মান N এ Initialize করতে হবে। একটি মানকে এক location নিচে বসিয়ে

Counter এর মান 1 বাড়াতে হবে। এভাবে 4 ও 5 নং Step চলবে। এরপর K তম অবস্থানে

ITEM Insert করব। নতুন Data insert হওয়ায় Size 1 বেড়ে গেল। তাই N = N + 1 করা হয়েছে।

Linear Array তে Data Delete বা বিয়োজনের Algorithm নিম্নরূপ ঃ

Algorithm:

LA	
A	0
B	1
C	2
D	3
E	4

Step 1: Start
Step 2: Set ITEM = LA [K]
Step 3: Repeat for J = K to N-1
    Move (J+1) element upward
    Set LA [J] = LA [J+1]
    (End of loop)
Step 4: Update N = N-1
Step 5: Stop

ব্যাখ্যা ঃ প্রথম কাজ হচ্ছে, যে উপাদানটি (ITEM) Delete করব তাকে আগে খুঁজে বের করতে হবে। ধরি, 2 location এর মান Linear Array (LA) হতে একবার করে উপরে উঠবে। এভাবে পার্যায়ক্রমে Looping এর শর্ত পর্যন্ত চলবে। একটি Data delete হবার পর মোট Data হতে একটি Data কমাতে N = N − 1 করা হয়েছে।