

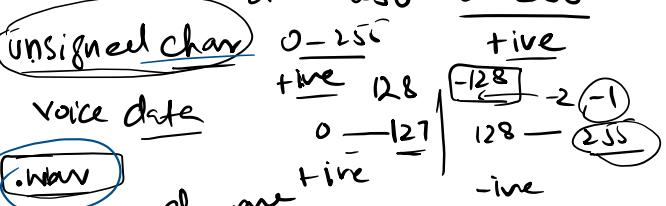
→ unsigned

Numbers
integer

int five
- five
unsigned int only five

Short
unsigned Short

char → 2^8 256 0-255



Voice data

.wav

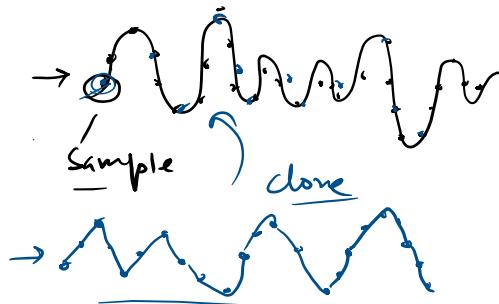
Sound wave
Signal Analog → digital

.wav
compressed
mp3 short data
.aac small

0, 1
digital

float

32-bit float



quality sampling rate

→ samples per sec

44000

quality

data size

1-D dynamic Arrays: → 2D dynamic Arrays

Array of pointers: static (on stack)

int * ptr[i]

int * arrptr[5];

int x = 20;

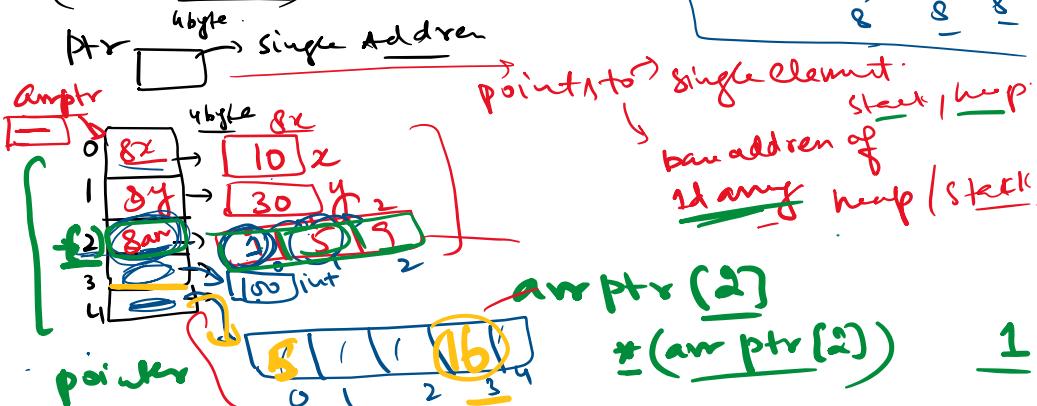
int y = 30;

arrptr[0] = &x;

arrptr[1] = &y;

int arr[3] = {1, 5, 9};

arrptr[2] = arr;



cout << *(arrptr[0]);

cout << *(arrptr[2]);

cout << arrptr[2][1];

cout << *(arrptr[2]+1)

10

1

5

5

1

5

9

* (arrptr[2]+0)

* (arrptr[2]+1)

* (arrptr[2]+2)

* (arrptr[4]+0) = 5;

arrptr[4][0] = 5;

(arrptr[4][3]) = 16;

Subscript

{ arrptr[2][0] } { 1 }
{ arrptr[2][1] } { 5 }
{ arrptr[2][2] } { 9 }

2D dynamic Arrays :

Dynamic Array:
1- An array of pointer \rightarrow stack \rightarrow partially dynamic 2D array -
 \rightarrow heap \rightarrow completely dynamic array.

2 → Store 1d array at each index of arrays of pointers.

④ const int s=5;
int * aptr [s]={0};

```
aptr[0] = new int[5];  
aptr[1] = new int[5];  
aptr[2] = new int[5];  
:  
:
```

loop [for(int i=0; i<8; itt)
→ aptx[i]=new int [5];]

\rightarrow [for (int i=0; i<s; i++)
 a.ptr[i]=new int [i+1];]

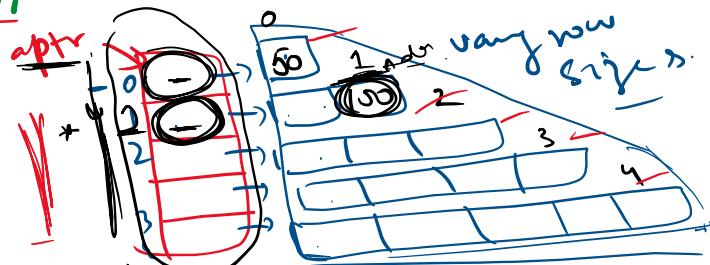
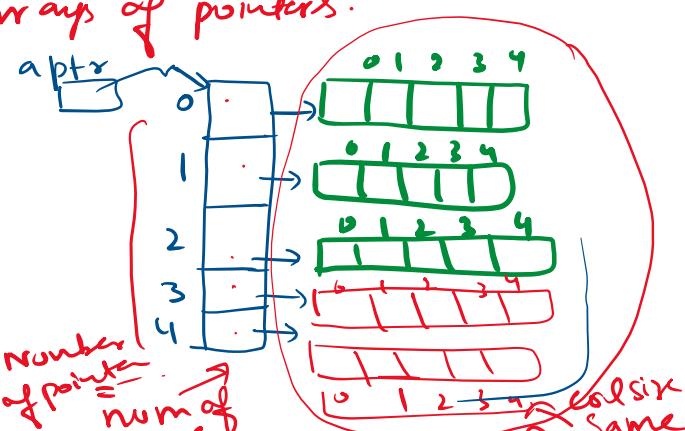
Yours,
Array of pointers
↓
array of date
↓
int
char

apts [0][0] = 50j

$\rightarrow \text{aptr}[\underline{0}][\underline{1}] = \underline{100}$

$$\rightarrow \text{aptr}[\underline{1}][\underline{1}] = 150j$$

$$\text{prece} \quad \text{ni pr} \\ \text{*(aptr + 1))} [\underline{1}] = \underline{100} \\ \text{var-1} \\ \text{arr of} \\ \text{pointer} \\ \text{elements} \\ \text{data}$$



$$*(\text{aptr}[\omega] + 0) = 50$$

$$X * (\underline{(ap + r + o)} + o) = So$$

$$\cancel{A} \sqrt{(a \cancel{\text{ptr}[1]} + \underline{1})} = \underline{100}$$

$$*(\underbrace{*(\text{a} \text{ptr} + 1)}_{\text{offset pointer}} + 1) = 100;$$

$$\boxed{*} (\text{apt} \times \underline{\underline{[0]}} + \underline{\underline{[0]}}) = \underline{\underline{[50]}}$$

2D Arrays

$$3 \times 5$$

```

    for (int i=0; i<s; i++)
        for (int j=0; j<s; j++)
            cin >> captr[i][j];
        cout << captr[i][j];
    }
}

```

unary \Rightarrow binary

52

A hand-drawn diagram illustrating two parallel wires. The top wire is labeled "now wire" and the bottom wire is labeled "color wire". Arrows above the wires indicate the direction of current flow.

disposal value

Deallocate Dynamic portion.

→ $\{ \text{delete } [] \text{ aptr}[0]; \\ \text{delete } [] \text{ aptr}[1]; \\ \quad \quad \quad \vdots \\ \text{delete } [] \text{ aptr}[n]; \}$ → $\left[\begin{array}{l} \text{for (int i=0; i<} n; i++) \\ \quad \quad \quad \underline{\text{delete } [] \text{ aptr}[i];} \end{array} \right]$

Arrays of pointers.

→ Compile time

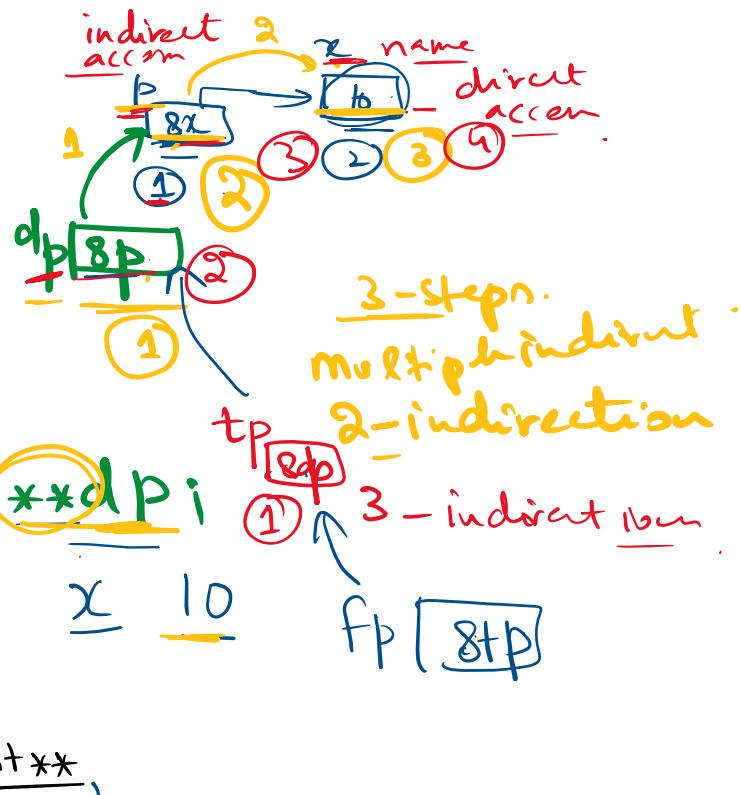
→ we cannot resize or change array of pointers. (Number of rows cannot be changed)

→ we can change columns → same size

move Array of pointers to heap:

→ multiple indirection: → (Store address of one pointer in another pointer)

$\text{int } x = 10;$
 $\text{int } * \text{ptx} = \underline{8x}; \text{ int }$
 $\text{cout} \ll \underline{x} \ll * \text{pi}$
direct indirect
 $\text{int } * \text{dp} = \underline{8pi}; \text{ int } *$
2d-pointer



$\text{cout} \ll \underline{dp} \ll * \underline{dp} \ll ** \underline{dp}; \text{ int }$

$x \underline{10}$

$\text{int } * \text{** } \underline{tp} = \underline{8dp}; \text{ int } **$

$\text{cout} \ll \underline{tp} \ll * \underline{tp} \ll ** \underline{tp} \ll *** \underline{tp}; \text{ int }$

$\text{int } * \text{*** } \underline{fp} = \underline{8tp}; \text{ int } ***$

$x \underline{10}$

$\text{cout} \ll \underline{fp} \ll * \underline{fp} \ll ** \underline{fp} \ll *** \underline{fp}; \text{ int }$

$x \underline{10}$