

void fun1(int x); void fun2(~~int~~ main());
 void fun1(~~int~~ * p); } int x = 10
 void fun3(~~int~~ * ~~* p~~) by reference; int * p = &x; fun(p); fun(&x);
 { ~~(p = nullptr)~~ ref. }
 void fun3(~~int~~ * ~~* p~~) by reference; fun3(p);
 { ~~(**dp = 100;~~
~~*dp = nullptr;~~ }
~~dp = nullptr;~~ val → ^{2D} higher dimension pointers.
 { ~~dp = nullptr;~~ }
 3 ~~dp = nullptr;~~ by value
 → 1 Dpointer
 → Data
 referenc

void fun3(8 p);
 int ~~** dp = 8pi~~
 fun3(dp);

void fun4(~~int~~ * ~~* 8 dp~~) ref (fun4(dp));
 { ~~(** dp = 100;~~
~~* dp = nullptr;~~ }
~~dp = nullptr;~~ ref

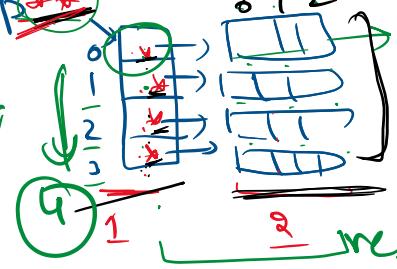
void fun5(8 dp)
 int ~~*** tp = 8dp;~~
 fun5(tp);

void fun5(int *** tp)
 { ~~*** tp = 50;~~
~~** tp = nullptr;~~
~~* tp = nullptr;~~ } value
 { ~~tp = nullptr;~~

void fun6(int *** 8 tp)

void fun7(int *** fp)
 { ~~*** fp = 4;~~
~~** fp = nullptr;~~
~~* fp = nullptr;~~
 { ~~fp = nullptr;~~

→ fun7(8 + p);
 int ~~*** fp = 8 tp;~~
 → fun7(fp);

1) Allocation
 void fun(int **~~dp~~ print~~st~~, int r, int c); ~~dp~~
 void fun(int **dp, int r, int c);
 void fun(int ***tp, int r, int c); 

 2) Deallocation
 void fun(int **dp, int r);
 void fun(int ***tp, int r); 

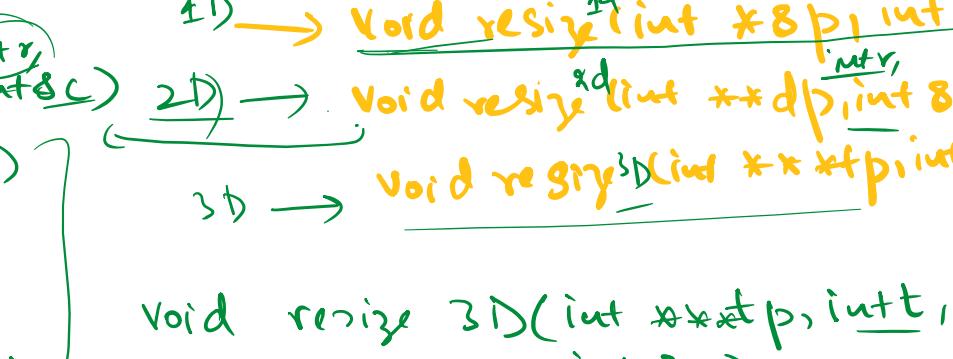
 3) Print data
 void fun(int **dp, int r, int c); 

 4) Resize columns
 void fun(const int **dp, int r, int c); 

 5) Resize rows
 void fun(int **dp, int r, int &c); fun(dp[0]);
 void fun(int **dp, int &r, int c);
 void fun(int ***tp, int &r, int c); 

 6) Whole 2D Array

6) Modify data [void fun(int **dp, int r, int c);]
 only

void resized2D(int **dp, int r, int c); 

 for(int i=0; i<r; i++)
 { resized2D(dp[i], c); }

1D → void resize1D(int *p, int c);
 2D → void resize2D(int **dp, int r, int c);
 3D → void resize3D(int ***tp, int r, int c)

void resize3D(int ***tp, int r, int c)
 int t

{
 for(int i=0; i<t; i++)
 resized2D(tp[i], r, c);
 }

1
 2