

C module PAPER-3

Q.1 main()

{ int i=40;

do

{ pf("y.i.d", i++);

} while (3,4,3,2,1,0,4,1);

return 0;

O/p = 40

3g

Q.2 main()

{ int a=10;

if (pf("y.a", a>=10)-10)

for(;;)

break;

else;

O/p = 1

3g

Q.3 vector printing times

main()

{ int x;

for( x=-1, &x<=10; x++ )

{ if(x<5)

continue;

else -

break;

pf("vector");

3g return 0;

O times  
vector  
printed

No O/p

3g

Q.4 minimum no. of iteration in do while 1

Q.5 auto int a=5

main()

Compile time  
error.

auto can  
be global

{  
3g

lenovo

Q.11 main()
$$\begin{array}{l} \downarrow \\ \text{int } i=300 \\ \text{char } *ptr=f^{\circ} ; \\ \downarrow \\ \text{ptr} = 2 \\ \text{printf("y-d",i) ;} \end{array}$$

1	0010 1100
256	

$$0/p = 556$$

Q.12 main()
$$\begin{array}{l} \downarrow \\ \text{int arr[2][3][2]=\{ \{ 2,2 \}, \{ 7,8 \}, \{ 3,4 \} \},} \\ \quad \{ \{ 2,2 \}, \{ 2,3 \}, \{ 3,4 \} \} ; \\ \text{printf("y-d", **(arr+1)+2+7),} \end{array}$$

$$0/p \Rightarrow \underline{16}$$

Q.13 main()
$$\begin{array}{l} \downarrow \\ \text{char thought[2][3]=\{ "Don't walk in front} \\ \quad \text{"of me", "I am} \\ \quad \text{"not follow"}, \\ \text{printf("y-c y-c", thought[0]+9), thought[1]+5)} \\ \downarrow \\ 0/p = K- \end{array}$$

(14) main()

$$\begin{array}{l} \downarrow \\ \text{int } i=257^{\circ} \\ \text{int } *iptr=f^{\circ} ; \\ \text{printf("y-d y-d", ((char)iptr), ((char)iptr+1))} \\ \downarrow \\ 0/p \text{ is } \underline{\underline{11}} \end{array}$$

(15) main()

$$\begin{array}{l} \downarrow \\ \text{static int a[5]=\{ 0,1,2,3,4 \};} \\ \text{int } *p[5]=\{ a, a+1, a+2, a+3, a+4 \} ; \\ \text{int } *ptr=p^{\circ} ; \\ \text{ptr} = 4^{\circ} \end{array}$$

```
pf("%d %d %d", p1-a, *p1-a);  
*++p1;  
pf("%d %d %d", p1-a, *p1-a, **p1);
```

{ } O/P  $\Rightarrow$  111222

(16) main()

```
{ void *vp;  
char ch='g', *cp="goofy";  
int j=20;  
vp=&ch;  
pf("%c", *(char *)vp);  
vp=&j;  
pf("%d", *(int *)vp);  
vp=&cp;  
pf("%s", (char *)vp+3);
```

{ }

O/P  $\Rightarrow$  goofg

(17) main()

```
{ char arr[5], flag;  
arr[0]='A';  
flag = ((a==*a) && (a==arr[0]));  
pf("%d", flag);
```

{ }

O/P  $\Rightarrow$  1

(18) main()

```
{ func(1);  
{  
func(int i)  
{ static char str[] = {one, two, three, four};  
pf("%s", str[i++]);  
}
```

O/P  $\Rightarrow$  two.

lenovo

segmentation  
fault.

68 271

(19) main()  
↳ static char \* s[] = {black, white, yellow, violet};  
char \* \* ptr[] = {s+3, s+2, s+1, s};

P = ptr;

\*ptr; → O/P → te  
Pf ("y.s", \*ptr);  
O/P = CK

(20) main()  
↳ int argc, i;  
P = argc - 3;  
char \* argv[3] = {abcde, fghi, klmno};  
while (++argv, --argc, --i);  
Pf ("y.s", \*argv); O/P = fghi klmno

(21) main()

↳ union a;  
int i;  
char c[2];  
union a v;  
v.i = 256;

Pf ("y.d y.d y.d", v.i, v.c[0]++, v.c[1]++);

(22) main()

enum bool {true, false};

if (true == (2 == 3))

Pf ("Vector");

else Pf ("Institute");

O/P  
⇒ Institute

(23) struct st  
{ int x;  
int y;  
};  
main()  
{ struct st \*p = &{10, 20};  
printf("%d %d", p->x, p->y);

O/p  $\Rightarrow$  Segmentation fault.

(24) main()  
{ union u  
{ struct  
{ char c[2];  
char ch[2];  
} s;  
};  
struct  
{ short int i;  
short int j;  
} st;  
{ u.s.c = '1' '2';  
printf("%d %d", u.st.i, u.st.j);

O/p  $\Rightarrow$  268 271

(25) main()  
# define big(a,b) a>b?a:b  
# define swap(a,b) temp=a; a=b; b=temp  
main()  
{  
int a=3, b=5, temp=0;  
if ((3+big(a,b))>b) swap(a,b);  
printf("%d %d", a, b);

O/p  $\Rightarrow$  50

ushitago  
filetime  
var

lenovo

20:01  
02-06-20

(31) main()

#define int char;

main()

int i=65;

pf("x.d", size of(i));

OP = 1

8 bytes

(32) ~~main()~~ #define float ptr float & f1, f2 & p1

#define int ptr int &

float fptr f1, f2;

int ptr p1, p2;

TOP  $\Rightarrow$  float, int

(33) char \*gc()

static char x[1024];

return x;

main()

char g1 = "first string";

strcpy(g1), g1);

g1 = g1;

strcpy(g1, "second string");

pf("x.s", g1));

TOP  $\Rightarrow$  Second string

(34) main()

pf("x.d", -2);

OP  $\Rightarrow$  Compile time error.

85

3  
Sun  
S  
U  
N  
gUn  
g  
U  
n

(44) `return inside { green, red, orange, blue, white }  
 main()  
 {  
 green = green + 1  
 p + ("%.d %.d", green, m);  
 }`

Compiletime errors

(45) `union { int x;  
 char y;  
 struct { char z;  
 char y;  
 int x;  
 } P;  
 } Q;`

size of Q is 8 bytes

(46) `main()  
 {  
 int a[] = {10, 20, 30, 40, 50};  
 char *p;  
 p = (char *) a;  
 }`

How to print 80

$\Rightarrow \boxed{\star(P+16)}$

$P + ("%.d", \star((int *)P) + 4))$

LENOVO

C paper

(1) main()

```

    int x=123;
    i = printf("C.%d", x);
    for (x=0; x<=i; x++)
        pf ("%d", x);
    return 0;
  
```

C++0123

(2)

```

for(i=0; i<5; i++)
    for(j=i; j>0; j--)
        x = i+j+1;
    pf("%d", x);
  
```

4+1+1=6

(3)

```

fun(int i){
    if (i>2)
        return 0;
    else
        return 1;
}

main()
{
    i=3;
    p = fun(i);
    q = fun(i);
    pf ("%d", p);
}
  
```

i=1

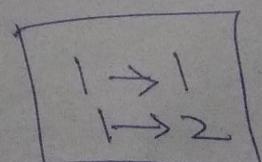
(4) Scope of a global variable which is declared as static within a file.

(5) void fun()

```

    int x=0
    static int y=0;
    x++, y++;
    pf ("%d->%d", x, y);
}

main()
{
    fun();
    fun();
}
  
```



```

⑦
int a = 5;
{
    int b = 10;
    ++b;
    ++a;
}
int a = 20;
++a;
a = ++b;
}
++a;
++b;
printf("%d %d", a, b);
}
printf("%d", a);

```

7, 13, 7

- ⑧ What is use of static function? → static int sum()  
 Security purpose  
 ↓  
 Not access to other file.
- ⑨ Even if integer / float arguments are supplied at command prompt → strings → ① integer  
 ② Double  
 ③ String
- ⑩ return type of calloc → void \*
- ⑪ malloc → stdlib.h

```

char *p = (char *) 0;
int *q = (int *) 0;
float *r = (float *) 0;
double *s = 0;
a = p + 1, b = q + 1, c = r + 1, d = s + 1

```

Ans → 1, 4, 4, 8

- ⑫ realloc → void \* realloc (void \*, size)  
 free → void free (void \*)

⑬ func(int a, int b)  
 {
 return(a == (a == b));
 }

main()

```

int process(); func();

```

printf("the Value is %d", process(func, 3, 6));

Ans = 0

LENOVO

```
Process (int (* printf)(c), int val1, int val2);
    ↳ return ((* printf)(val1, val2))
```

28

Ans →  Call back function

(14)

char \*g(c)

{ static char x[1024];  
return x; }

main()

{ char \*g1 = "first string";  
strcpy(g1, g1); }

g1 = g();

strcpy(g, "second string");

printf("%s", g());

29

Ans

Second String

(15)

main()

{ int i, argc;

i = argc = 3;

char \*argv[3] = {"abcde", "fghi", "klmn"}

char \*argv = argv;

while (++argv, --argc, --i)

printf("%s", \*argv);

30

Ans → fghi klmn.

(16)

main()

{ union a

{ int i;

char c[2]; };

31

union a {

int i = 256;

printf("%d %d %d", i, i,

i - C[0] +

i - C[1] + )

Ans → 513, 0, 1

Q16 enum mode { green, red, orange, blue, white }  
green = green+1

Ans → Compile time error.

Q17 enum x { false, true };

```
main()
{ int i=1;
  do
  { pf("%d", i);
    i++;
    if (i<15)
      continue;
  } while (false);
```

Ans = 1

Q18 if we or more mb → nested structure.

Q19 main()

```
{ enum bool { True, False };
  if (True == (2==3))
    pf("%vector");
  else
    pf("institute_!!");
```

Ans → ...Vector

Q20 main()

```
{ struct xx
  { int x=3;
    char name[]="hello";
  };
}
```

error  
we can't  
initialize inside  
structure.

Q21 #define a 10

main()

```
{ pf ("%d---", a);
```

foo()

```
pf ("%d.", a);
```

void foo()

```
{ # undef a;
  # define a 50
}
```

Ans  
10 10

Q22 #define big(a,b) a+b

#define swap(a,b)

temp=a;

a=b;

b=temp;

void main()

```
{ int a=3, b=5, temp=0
```

```
if ((3+big(a,b))>b swap(a,b))
```

```
pf ("%d %d", a, b);
```

}

Ans → 5,0

will be the output of the program?

20:08  
02-06-2016

lenovo

(25) A default setting enumerated start from 0:

#define Cal (a,b)(a+b)/a-b)  
void main()

{ int a=20, b=10;

PF ("Y.d", Cal (a+4, b-2));

Ans → [4]

}

(26) #define MAN(x,y) ((x)>(y))?(x):(y);

main()

{ int i=10, j=5, k=0;

k=MAN (++i, j++);

PF ("Y.d", Y.d Y.d", i, j, k);

} return 0;

Ans → [12 6 12]

(27) #define f(g1,g2) g##g2

main()

{ int var12=100;

PF ("Y.d", f (var, 12));

O/P → [100]

}

(28) Source text → "Be my friend\n".

main()

{ FILE \*fs, \*ft;

char C[10];

fs=fopen("source text", "r");

C[0]=getchar(fs);

fseek(fs, 0, SEEK\_END);

fseek(fs, -3L, SEEK\_CUR);

fgets(C, 5, fs);

putchar(C);

return 0;

Ans → [nd]

(30) The fopen() return 0

(31) main()

{ FILE \*fp;

PF ("Y.d", "++EOF"));

}

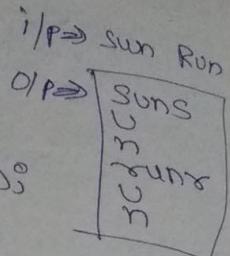
O/P → [Error]

++(-1)

A ↑

can't

(32) main ( int argc, char \* argv[] )  
` if (argc == 1)  
` pf ("error");  
` exit (1);  
` while (\*argv, --argc)  
` pf ("%s", \*++argv);  
` while (argc, ++argv)  
` pf ("%s\n", \*(argv)++);  
` putchar ('\n'); } }



(33) mention 3 facilities provided by a preprocessor.

- Include header files
- Replace macro
- Conditional compilation

(34) Va-list is predefined datatype.  
in stdarg.h.

(35) main()  
` int i = -1;  
` +i;  
` pf ("%d %d", i, +i);  
` }

Ans → [-1, -1]

(36) ternary operator.

$x > y ? x : y$

(37) size of p1 & p2?

Struct X { int ; char };

Ans [4, 4]

lenovo

(38) main()

```

int x=10;
printf("%d", x);

```

Ans  
 $10/p = 10$

(39)
 ~~#include < iostream.h >~~  
 #include < > & #include " "

predefined  
 user defined

(40) unary operator
sizeof, ++, ~, !, --, &
(41) main()

```

static char arr[] = {"black", "white",
                     "yellow", "video"};
char *ptr[] = {arr+3, arr+2, arr+1, arr};
ptr++;
printf("%s", *ptr++);

```

Ans  $\Rightarrow$  te

(42) main()

```

float a=123.4, b=456.8
a^=b^=a^=b;
printf("%d %d", a, b);

```

Ans  $\Rightarrow$  error.

bitwise can  
 operate  
 on  
 float.

(43) main()

```

int i;
for(i=0; i<5; i++)
    int i=0
    printf("%d", i);
    i++;

```

Ans  $\Rightarrow$  10 10101010

(44) main()

```

int n=10;
int func(int);
printf("%d", func(n));
int func(int n)
    if(n>0)
        return (n+func(n-1));
    else
        return 0;

```

Ans = 30

- ① → double-byte (Paper 2) (cont)
- ② → main is not a keyword ↗ Paper 2
- ④ break should be used in loops & switch cases.  
→ compile time error ↗
- ⑤ ~~#include<stdio.h>~~  
main()  
{ int const x=0;  
switch (5/4/3)  
{ case x:  
 case x+1:  
 case x+2:  
 default : printf("8nn");  
}
- Error: label should be constant  
not variable ? ↗
- ⑥ label always in ascending order  
in switch cases  
false
- ⑦ main()  
{ int i=1, j  
switch (i)  
{ case 1 : Pf("Hello");  
case 2 : Pf("Hi");  
default : break;  
}
- O/P ⇒ Hello Hi
- 
- ⑧ main()  
{ int a=9;  
if (a=5)  
 Pf("It is Imprtant");  
else  
 Pf("It is not imp");  
}
- O/P ⇒ It is important
- 
- ⑨ ~~main()~~  
{ int x=8;  
if (x>4)  
 Pf("Veetan")  
else  
 Pf("the missing");  
}
- O/P ⇒ the missing
- 
- ⑩ main()  
{ int i,j;  
i=j=2,3;  
while (--i && j++)  
 Pf("ydyd", i, j);  
}
- return  
O/P ⇒ 1,3
- 
- ⑪ ~~main()~~  
{ int x=123;  
int p=pf("%d %d", x);  
for (x=0; x<=1; x++)  
 Pf("%d", x);  
return 0;
- C++  
O 1 2 3
- 
- ⑫ How many times  
the loop iterates?  
for (i=0; i=10; i+=2)  
 Pf("Hi")  
O/P ⇒ Hi will  
print infinite time

(13) How many time Vector prints

```

main()
{
    int x;
    for (x = -1; x <= 10; x++)
        if (x < 5)
            continue;
        else
            break;
    printf("Vector");
}
    
```

O/p  $\Rightarrow$  0 Times

(14) main()

```

int i;
for (i = 0; i < 5; i++)
    int i = 10;
    printf("%d", i);
    i++;
}
    
```

O/p  $\Rightarrow$  10 10 10 10 10

(15) main()

```

int i = 0;
char ch = 'A';
do
    Putchar(ch);
    while (i++ < 5 || ch <= 'F');
}
    
```

O/p  $\Rightarrow$  AAAAAABCDEF

(16) main()

```

int a = 6, b = 4;
while (a + b)
    printf("a=%d b=%d", a, b);
    a = a / 2;
    b = b / 2;
}
    
```

O/p  $\Rightarrow$

6	4
3	1
1	1
0	1
0	1
---	
---	

(17) main()

```

int i, j, x = 0;
for (i = 0; i < 5; i++)
    x = i + j + 1;
    printf("x=%d", x);
}
    
```

O/p  $\Rightarrow$  6

(18) main()

```

int x=3, counter = 0;
while ((x-1))
    & ++counter;
    x--;
    ↴
    printf("%d", counter);
    ↴ OP⇒ 2

```

(20) int check (static int  
 ↑  
 static,int);  
error  
Can't use static  
as formal argument

(21) main()

```

register a,b,x;
scanf("%d %d", &a, &b);
x=a+b;
error, registers address  
can't be used

```

(22) Every function must return a value  
→ false

(23) main()

```

static char s[25] = "Vector";
int i=0;
char ch;
ch = s[i]; printf("%c", ch);
ch = s[i+1]; printf("%c", ch);
ch = i++[s]; printf("%c", ch);
ch = ++i[s]; printf("%c", ch);
    ↴ OP⇒ e e c u

```

(24) int main()
int i, a[5]={2,4,6,8,10};
change(a,5);
for (i=0; i<=4; i++)
 printf("%d ", a[i]);
return 0;
void change (int \*b, int n)
int i;
for (i=0; i<n; i++)
 \*(b+i) = \*(b+i)+5;
OP⇒ 7, 15, 6, 8, 10

(22) A function assumes  
int data type as  
return type, if no  
return is specified in  
function prototype.

(23) A function type declared  
as char →  
actually return.  
char ⇒ ASCII value

(26) main()
   
 { int x;
   
 x=3;
   
 fx();
   
 pf("MAIN");
   
 }

fx()
   
 { int n;
   
 pf("f");
   
 if(n==0)
   
 fcn();
   
 }

FFFF MAIN

(27) auto int i=5;
   
 static int a=i+5;
   
 pf("%d", a);
   
 }

error

(28) Register variable are treated as static variable if CPU registers are not free.  
 → false  
 fop → auto

(29) Scope of a global variable which is declared static is within the file.

(30) void fn(int, int);
   
 main()
   
 { int a=5;
   
 pf("In main: %d %d", a++, ++a);
   
 fn(a, a++);
   
 }
   
 void fn(int a, int b)
   
 pf("fn: %d %d", a, b);

O/p → 6 7

8 7

(31) void main()
   
 { f();
   
 f();
   
 }
   
 f()

static int i=1;
   
 pf("%d", i++);
   
 i=--i, i++, i=i-2;
   
 i=(i++, i=i-2);

O/p → 1 -

(32) register help in faster execution.

(33) int fn(int v)  
if ( $v == 1$  ||  $v == 0$ )  
return 0;  
if ( $v \% 2 == 0$ )  
return fn( $v/2$ ) + 2;  
else  
return fn( $v - 1$ ) + 3;

O/P = 11

(34) void main()

int i=3, \*j, \*\*k;  
j=&i;  
k=&j;  
Pf(" %d %d %d", \*j, \*k, \*\*k);

O/P  $\Rightarrow$  333

(36) main()

int i=300;  
char \*ptr=&i;  
++ptr=2;  
Pf(" %d", i);

(37) main()

int a,b,c,d;  
char \*p=(char\*)0;  
int \*q=(int\*)0;  
float \*r=(float\*)0;  
double \*s=0;

1448

(38) main()

char \*pb;  
char mystring []="abcdefghijkl";  
ptr=mystring;  
ptr+=5;  
Pf(" %s", ptr);

O/P  $\Rightarrow$  fg

(39) int cap(int);  
main()

int n; [n=6]  
n=cap(6);  
Pf(" %d", n);  
int cap(int n)  
if (n<=1)  
return 1;  
return (cap(n-3)+cap(n))

lenovo

- (40) the invalid pointer ~~operator~~ arithmetic  
is all of them add, mul, div
- (41) what is the diff b/w the following  
~~data is constant~~ Const int \*ptr  
int \*Const ptr  
Const int &Const ptr.
- (42) when the digit 0 is enter from keyword  
what is send to

48

- (43) label in switch Case should not  
be float. → error.

(44) main()  
{ int ch=1;  
switch(ch)  
{ case 1: pf ("It is working\n");  
case 2: pf ("It's not working\n");  
vector<int> v; } }

Death

- (45) #include ~~int~~ a=5 ← error  
int main()  
{  
}  
auto variable  
should not  
declare as global

(46) main()  
{ char str[5] = "Genius";  
print(str);  
print(~~char~~ &s);  
if (&s)  
{ pf (str++); }  
pf ("-%c", \*s); }

execute  
continuously  
Segmentation fault

(4) main()  
 {  
 int q=3;  
 static int p;  
 if(p==0){cout<<"abc";}  
 else{q++;cout<<q;}  
 default:{break;};  
 }  
 O/P: abc

(5) int Counter(int i)  
 { static int count=0;  
 count=Count+1;  
 return(count);  
 }  
 main()  
 {  
 int i,j;  
 for(i=0;i<5;i++)  
 j=Counter(i);  
 cout<<j;
 }  
 O/P = 15

(6) static → scope of global  
 variable declared  
 in —

(7) #include <stdio.h>  
 void main()  
 { int i=3, j++ , k;  
 j = &i;  
 k = &j;  
 cout<<i<<j<<k;
 }  
 O/P = 3 3 3

(8) strings

(9) realloc

(10) main()  
 { char \*p="Aman";  
 cout<<p;
 }  
 O/P → Aman

(11) void fun (void \*p);  
 int p;  
 main()  
 { void \*vptr;  
 vptr=&p;  
 fun(vptr);  
 return 0;
 }  
 void fun (void \*p)  
 { int \*q;  
 q=(int \*)p;  
 p=(int \*)(&q[1]);
 }  
 O/P → 0

(12) int fun(char \*str)  
 { char \*ptr= str;  
 while(\*ptr++)  
 return ptr-str-1;
 }  
 void main()  
 { cout<<fun("vector");
 }  
 O/P = 0

(13) main()

{ char thought [2][30]  
 = { "Don't walk  
 in front of me--"  
 , "I am not  
 follow" };  
 cout<<thought[0][0];  
 cout<<thought[0][1];  
 cout<<thought[1][0];
 }  
 O/P → K-

the output of the program?

20:09  
02-06-2016

(15) main()  
` int i=257  
` int \*iptr=&i;  
` printf("%d %d %d", \*(char \*)iptr,  
` \*(char \*)iptr+1);

O/P  $\Rightarrow$ 

1	1
---	---

(16) main()  
` static int arr={10,1,2,3,4};  
` int \*ptr=a, a+1, a+2, a+3, a+4;  
` int \*p=ptr;  
` p++;  
` printf("%d %d %d %d", ptr-p, \*(ptr-a), \*(ptr+1));  
` \*++ptr;  
` printf("%d %d %d %d", ptr-p, \*(ptr-a), \*(ptr+1));

O/P  $\Rightarrow$ 

1	1
2	2

(17) main()  
` void \*NP;  
` char ch='g', \*cp="goofy";  
` int j=20;  
` \*p=&ch;  
` printf("%c", \*(char \*)NP);  
` NP=&j;  
` printf("%d", (int \*)NP);  
` NP=cp;  
` printf("%s", (char \*)NP+3);

O/P  $\Rightarrow$ 

g	oofy
---	------

(18) main()  
` int a,b,c,d;  
` char \*p=(char \*)0;  
` int \*q=(int \*)0;  
` float \*r=(float \*)0;  
` double \*s=0;  
` a=(int)(p+1);  
` b=(int)(q+1);  
` c=(int)(r+1);  
` d=(int)(s+1);

1448
------

- (19) main()
 

```
char a[5][5], flag;
a[0][0] = 'A';
flag = (ca == *a) && (*a == a[0][0]);
printf("%d", flag);
```

 $\boxed{\text{O/P} = 1}$
- (20) main()
 

```
union a
  {
    int i;
    char c[2];
  };
union a;
u.i = 256;
printf("%d %d %d %d", u.i, u.c[0]+u.c[1]+u.c[2]+u.c[3]);
```

 $\boxed{\text{O/P} = 513 \ 0 \ 1}$
- (21) structure st can't take the structure st itself if ~~as~~ as its member
   
TRUE
- (22) enum( false, True)
 main()
 

```
int i=1;
do
{
  printf("%d", i);
  i++;
  if(i<15)
    continue;
} while (false);
```

 $\boxed{\text{O/P} = 1}$
- (23) enum mode { green, red, grey };
 Green=Green+1;
 CORR.
- (24) Nested Structure
- (25) main()
 

```
enum bool (true, false);
O/P = method.
```
- (26) main()
 

```
struct xx; // error
int x=3;
char name[]="Hello", can't
            be
            seen
            from
            top
            level
```

 $\boxed{\text{O/P} = 0}$
- (27) main()
 

```
union {
  struct {
    char c[2];
    char ch[2];
  } s;
  struct {
    short int;
    short int;
  } t;
} x={12,1,15,1};
printf("%d %d", x.s.c[0], x.t.i);
```

 $\boxed{\text{O/P} = 268 \ 271}$

(29) `#define big(a,b) a>b ? a : b  
#define swap(a,b) temp=a  
                a=b  
                b=temp.`  
`void main() {  
    int a=3, b=5, temp=0;  
    if((3+big(a,b))>b) swap(a,b);  
    printf("%d %d", a, b);  
}`

$\boxed{OIP \Rightarrow 50}$

(30) in macro call the control  
is passed to the macro  
 $\rightarrow$  false

(31) macros have local scope  
 $\rightarrow$  false

(32) what is type of variable  
f2 & P2

`#define floatptr float*`  
`#define intptr int*`  
`floatptr f1, f2;`  
`intptr p1, p2;`

$\rightarrow \boxed{\text{float, int}}$

(33) `#define sum(a,b,c) a+b+c  
#define avg(a,b,c) sum(a,b,c)/3  
#define geo(a,b,c) avg(a,b,c)>=60  
#define lee(a,b,c) avg(a,b,c)<=60  
#define des(a,b,c,d)  
    { d==1? geo(a,b,c):  
      lee(a,b,c)}`

main()  
`{ int num=70; OIP  $\Rightarrow$  geo  
    char ch='0';  
    float f=2.0;  
    if(des(num, ch, 0)) puts("des");  
    else puts("geo"); }`

$\boxed{OIP \neq geo}$

(34) `int main() {  
    FILE *fp;  
    char ch, str[7];  
    fp=fopen("try.c", "r");  
    fseek(fp, 9L, SEEK_CUR);  
    fgets(str, 5, fp);  
    puts(str);  
    return 0; }`

$\boxed{OIP \Rightarrow \text{agpu}}$

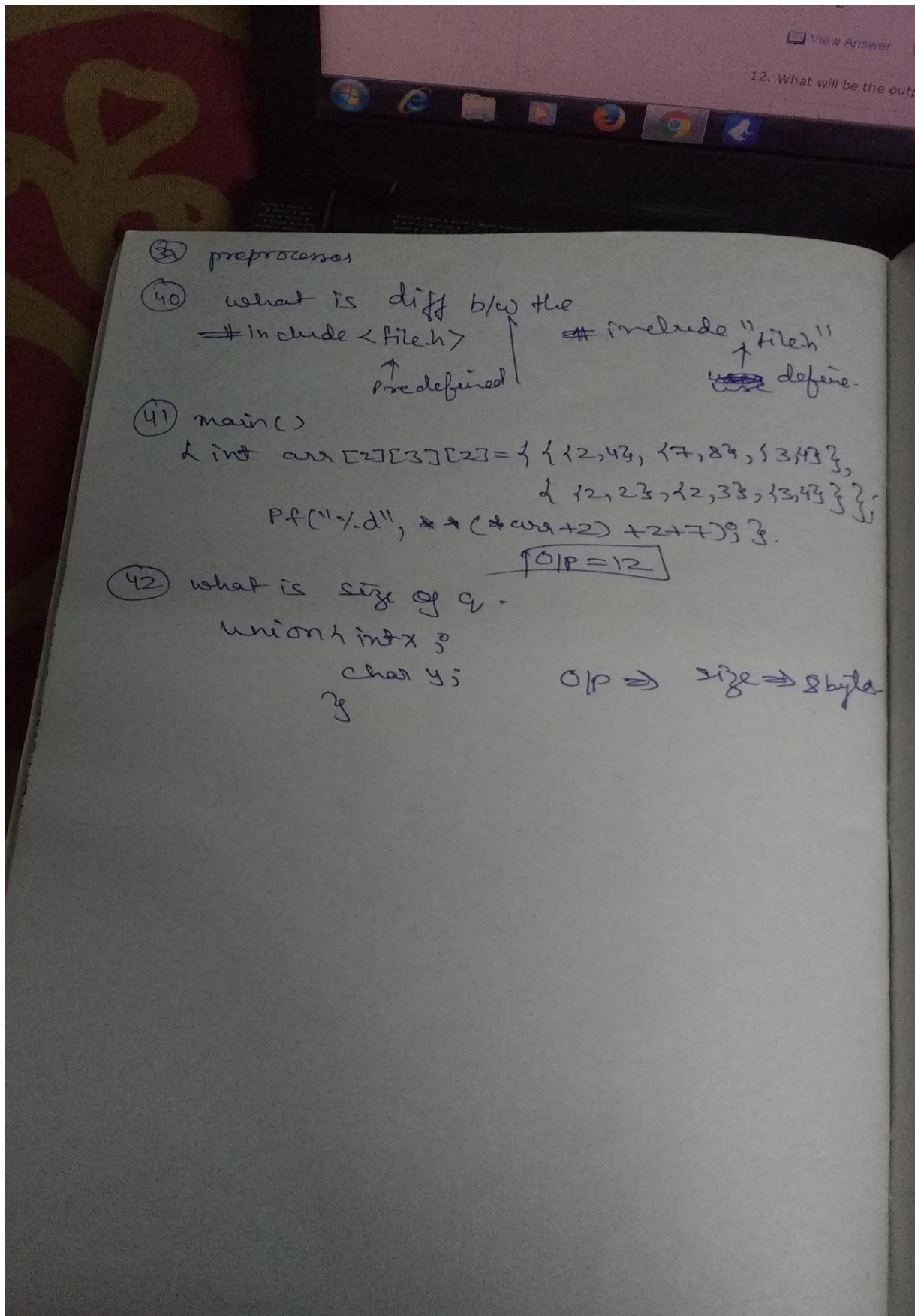
(35) function call  
`fseek(fp, 0, 0)`  
 $\rightarrow$  same  
`rewind(fp);`

(36) `main() {  
    FILE *fp;  
    Pf("Y.d", ++EOF);  
}`

$\boxed{\text{error}}$

(37) `textfile=vector  
main() {  
    file *fp;  
    ptr=fopen("file.txt", "r");  
    Pf("/c", fgetc(ptr));  
    fseek(ptr, 2, 1);  
    Pf("/d", fgetc(ptr));  
    fseek(ptr, -2, 1);  
}`

$\boxed{OIP \neq VTC}$



## PAPER - 2

1

main()

{ int a=2 , x=10 ;

    if (a==2)

        if (x==8)

            printf("Hello")

(Hi)

        printf("Hi")

2

{ int i=1 , j=9 ;

    if (i>=5 && j<5)

        i = i+2

    printf("Y.d", i);

3

{ int x=3

    if (x==2)

        x=0;

    if (x==3)

        x++;

    else x+=2;

}

    printf("Y.d", x);

(2)

4    { int a=off;  
      if(a<<4>>12)  
         pf("Right")  
      else  
         pf("Wrong");  
     }

wrong

if(0)  
else  
wrong

5    main() { int i,j;  
      i=j=2,3;  
      while(-i && j++)  
         pf("x.d,y.d",i,j);  
      return 0;  
     }

1,3

6    The minimum no. of times while loop is  
      executed is ?  
     → zero

7    { int x=0;  
      for(x=1 ; x<4 ; x++)  
         pf("x=yd",x);  
     }

4

8    main()  
     void \*VP;  
     Char ch='J', \*CP="JACK";  
     int j=65,  
         VP=&ch;  
     pf("x-d",\*(char\*)VP);      J  
         VP=&j;

JACK

Pf ("x-C", \*(int\*)VP), ← A  
VP = CP;  
Pf ("x-S", (char\*)VP+2), ← CR  
return 0;

}

9

main ()  
└ Pf ("x-C", "abcde");  
return 0;

}

(a)

10 Array name represents the base address of the starting element.

11 If size of an array is less than the number of initializer, the size grow

→ false

(✓)

12

main ()  
└ char stop[] = "Never ever say No";

char \*chp, c = 'e';

int i, j;

chp = Strchr (stop, c);

i = chp - stop

for (j = 0; j <= i; j++)

Pf ("x-C", stop[j]);

}

Never ever

(✓)

3 main()  
2 int buf[5]  
int \*bptr= &buf[3];  
bptr[-1]=2; bptr[-2]=1;  
bptr[3]=0;  
pf ("x.d", buf[2]);

(2)

}

4 main()  
2 int A[5]={1,2,3,4,5};  
int x;  
x= \*A+1 - \*A+3;  
pf ("y.d", x);

X  
= 4

1+1-1+3 = 4

}

5 main()  
2 char \*p="Vector institute";  
\*p++;  
pf ("x.S", p);

vector institute

p++;

ctor institute

}

6 ~~#include~~ auto int a=5; // Compile time error  
int main()  
2 int x;  
x=a+a&a+a<a;  
pf ("x.d", x);

X

17 main()

```
{ int a[5] = {2,4,6,8,10};  
    change(a,5);  
    for(i=0; i<=4; i++)  
        pf("xd", a[i]);
```



void change (int \*b, int n)

```
{ int i;  
    for(i=0; i<n; i++)  
        *(b+i) = *(b+i) + 5;
```

ans  
2, 13, 6, 8, 10

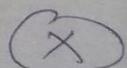
18 #include <stdio.h>

```
int check (static int, static int);
```

int main ()

Compile time  
error

we can't



take static in function.

19

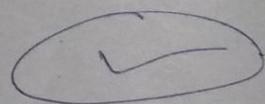
```
int a=5;
```

main()

```
{ int x;
```

```
x = !a + change();
```

```
{ pf("xd", x);
```



int change()

```
{ a=0;
```

```
{ pf("xd", a);
```



integer data as a structure

2) clear function must return  
false

3) func(int i)

& if(i>2)  
return 0;  
else  
return 1;

①

~~(x)~~

4) main()

& int i=3;  
i = func(i);  
j = func(i);  
pf (" %d ", i);

5

5 162421  
0 0 1 1 1 0

6) main()

static char s[25] = "vector";

int i=0;

char i=0;

char ch;

ch = s[i++]; pf (" %c ", ch);

ch = s[i++]; pf (" %c ", ch);

ch = i++[s]; pf (" %c ", ch);

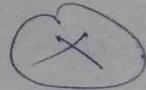
ch = i++[s]; pf (" %c ", ch);

6

vector

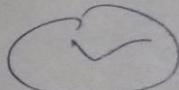
24 main()  
{ int x;  
x=3;  
f(x);  
printf("MAIN");  
}  
f(int n)  
{ printf("F");  
if(n==0)  
f(n-1);  
}

FFFFMAIN



25 what is default storage class of global variable.

B extern

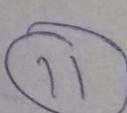


26 which storage class may help in faster execution?  
→ Registers.



27 int fn(int v)

{ if(v==1 || v==0)  
return 1;



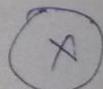
if (v>2==0)  
return fn(v/2)+2;

else

return fn(v-1)+3;

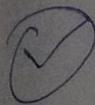
main() { printf("%d", fn(7)); }

3



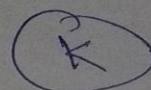
28 main()

& char thought[2][3] = { "Don't walk in front of  
me - .", "I am not followed" };



Pf ("y.c.y.c", \*(thought[0]+9), \*(thought[0]+  
+5)),

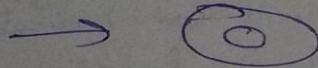
y →



+5)),

29 what is the value of a[0][2] in int

a[3][4] = {{1,2,3,4,5}, {1,2,3,4,5}, {1,2,3,4,5}, {1,2,3,4,5}};



30 main()

& static int a[] = {0,1,2,3,4};

int \*p[] = {a,a+1,a+2,a+3,a+4};

int \*ptr = p;

ptr++;

Pf ("y.d.y.d.y.d", ptr-p, \*ptr-a, \*ptr);

\*ptr++,

Pf ("y.d.y.d.y.d", ptr-p, \*ptr-a, \*ptr-a,  
, \*ptr);

y ⇒

1	1	1
2	2	2

31 main()

& char a[5][5], flag;

O/P ⇒ ①

a[0][0] = 'A';

flag = ((a == \*q) && (\*a == q[0]));

y Pf ("y.d", flag);

main()

& char \*ptr;

char mystring [] = "abcdefghijkl";

ptr = mystring ;

ptr += 5;

pf ("%s", ptr);

fg

33

mention atleast

→ next do calculation

size of ( )

→ works on

any datatype

→ it will include

'\0'

→ size of operator

three difference b/w strcpy()

→ do calculation

strlenc()

& strlent()

→ work for only string

→ it will not include '\0'

→ strlent is function.

34 write the prototypes for 1) strcpy ()

2) strcat ()

3) strlent ()

char \*

strcpy (char \*, char \*)

→ strcat (char \*, char \*)

→ int strlent (char \*)

35

main()

& int arr [5] = {10, 20, 30, 40, 50};

pf ("%d\n", &arr[4] - &arr[0]);

3

4

36 The invalid pointers arithmetic is care  
→ all are invalid

37 All the string manipulation function have been included  
in the String.h header.

38 inter linkage & external linkage

39 a parameters of

40 int cap(<sup>char</sup> s)  
main  
{  
 n=cap(s)  
 pf(" %d", n);  
}  
int cap(int n)  
{  
 if (n<=1) return 1;  
 else return (cap(n-1)+cap(n-1));  
}

41 main()  
{  
 int i=20, k=0;  
 for (j=1; j<1; j=1+4\*<sup>int</sup>(j))  
 k+=j<10?4:3;  
 pf(" %d", k);  
}

OP  $\Rightarrow$  6

42 main() { char str[2] = "G"; printf("%s", str); }

Print  
↳ Print(str);

(X)

Print  
↳ Print(~~char \*s~~)

Segmentation fault

↳ if (\*s) Print(s++);

↳ pf ("%c", \*s);

43 How print() using P?

main()

{ int a[] = {10, 20, 30, 40, 50}; }

char \*P;

P = (char \*) a;

↳

Ans →

pf ("%d", \*(P+i));

or  
pf ("%d", \*(int \*) P + i);

or  
pf (short \*) P + i;

44 #include <stdio.h>  
int main()

{ int i;

for (i=0; i<5; i++)

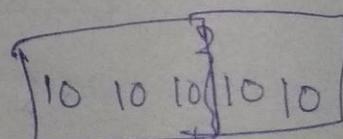
{ int i=10;

pf ("%d", i);

i++;

return 0;

(X)



45 main( )

{ int n=10 ;

int fun(int);

printf("%d", fun(n));

}  
int fun(int n)

{ if (n>0)

return (n+fun(n-2));

> else return 0;

}

of 30

(30)

return (10+ fun(8))

8+12 return (8+ fun(6))

6+6 return (6+ fun(4))

4+2 return (4+ fun(2))

2 return (2+ fun(0))