**C - Aptitude Questions**

1) #include<stdio.h>

main()

{

char c=125;

c+=10;

printf("%d\n",c);

}

2) #include<stdio.h>

main()

{

int c=10;

printf("%d %d %d %d\n",c++,++c,c++,++c);

}

3) #include<stdio.h>

main()

{

int n=0xfffd;

printf("%x\n",n & 1);

}

4) #include<stdio.h>

main()

{

unsigned char temp=0xff;

printf("\n%d\n",temp >> 5);

printf("\n\r %d\n",temp >> 5);

printf("\n%d\n",temp);

}

5) Which is predominant; precendece or associativity?

6) Write a Program to swap two variables without using a temporary variable.

7) #include<stdio.h>

main()

{

unsigned int i=13;

int count=0;

for(;i;i>>=1)

{

if(i & 1)

{

count++;

}

}

printf("%d\n",count);

}

8) #include<stdio.h>

main()

{

int a=2;

if(a==2)

{

a=~a+2<<1;

printf("%d\n",a);

}

}

9) #include<stdio.h>

main()

{

int i=258;

char c1=i;

char c2=i>>8;

printf("c1=%d c2=%d c1+c2=%d\n",c1,c2,c1+c2);

}

10) #include<stdio.h>

main()

{

int a=-12,b=-12,c=-12,d=-12;

a=a>>1;

b=b>>2;

c=c>>3;

d=d>>4;

printf("%d %d %d %d\n",a,b,c,d);

}

11) #include<stdio.h>

main()

{

float a=0.55,b=0.80;

if((a && b)>0.80)

printf("if\n");

else

printf("else\n");

}

12) #include<stdio.h>

int y=10;

main()

{

int x=10;

printf("%d %d %d\n",x=30,x>10,x=10);

printf("%d %d %d\n",y=30,y>10,y=10);

}

13) #include<stdio.h>

main()

{

int i=-3,j=2,k=0,m;

m=++i || ++j && ++k;

printf("%d %d %d %d\n",i,j,k,m);

}

a) -2 2 0 1

b) -2 3 1 0

c) -2 2 1 1

d) none of these

14) #include<stdio.h>

main()

{

int i=3;

int j;

j=sizeof(++i + ++i + ++i);

printf("i=%d j=%d\n",i,j);

}

15) #include<stdio.h>

main()

{

int a,b,c,d;

a=3;

b=5;

c=a,b;

d=(a,b);

printf("%d %d\n",c,d);

}

16) #include<stdio.h>

main()

{

int a=0,b;

b=(a=0)?2:3;

}

17) #include<stdio.h>

main()

{

int a=5,b=6;

int max;

a>b?max=a:max=b;

printf("max=%d\n",max);

}

18) #include<stdio.h>

main()

{

int i=4;

switch(i)

{

default: ;

case 3:

i+=5;

if(i==8)

{

i++;

if(i==9)

break;

i\*=2;

}

i-=4;

break;

case 8:

i+=5;

break;

}

printf("i=%d\n",i);

}

19) #include<stdio.h>

main()

{

int q,i,j,count;

i=j=0;

q=2;

count=6;

switch(3)

{

case 0:while(--count>0)

{

case 1:++j;

case 2:++i;

case 3: ;

case 4: ;

case 5: ;

}

printf("%d ",i);

printf("%d \n",j);

}

}

20) #include<stdio.h>

#define MAX 100

#define MIN 100

main()

{

int x=200;

if(x>MAX)

x=1;

else if(x<MIN)

x=0;x=50;

}

a) 200

b) 1

c) 0

d) 50

21) #include<stdio.h>

main()

{

int i;

for(i=4;i<10;i=i+2)

{

i=i-2;

printf("%d ",i);

}

}

22) For the below code how many times loop will be rotate

for(i=0;i==10;i+=2)

printf("HI...\n");

a) 10

b) 2

c) 5

d) none of these

23) #include<stdio.h>

main()

{

unsigned int i=5,c;

for(c=0;i;c++)

i&=i-1;

printf("%d\n",c);

}

a) finds out if i is even or odd

b) finds out the number of bits set in i

c) finds out the number of bits non set in i

d) none of these

24) #include<stdio.h>

main()

{

int i,j,c1=0,c2=0;

for(i=0;i<10;i++,c1++)

for(j=0;j<20;j++,c1++);

for(i=0;i<20;i++,c2++)

for(j=0;j<10;j++,c2++);

printf("c1=%d c2=%d\n",c1,c2);

if(c1==c2)

printf("c1==c2\n");

else if(c1>c2)

printf("c1>c2\n");

else if(c1<c2)

printf("c1<c2\n");

else

printf("not \n");

}

25) #include<stdio.h>

main()

{

int i=20,j,k=0;

for(j=1;j<i;j=j+4\*(i/j))

k+=j<10?4:3;

printf("%d\n",k);

}

a) 4

b) 0

c) 1

d) 2

26) #include<stdio.h>

main()

{

int a[5]={2,3};

printf("%d %d %d\n",a[2],a[3],a[4]);

}

a) garbage

b) 2 3 3

c) 3 2 2

d) 0 0 0

27) What is the difference between array and pointer.

28) #include<stdio.h>

main()

{

char \*a,\*b;

a=b=NULL;

a++;

b++;

b++;

// printf("%d\n",a+b);

printf("%d\n",a-b);

}

29) #include<stdio.h>

main()

{

int arr[]={12,13,14,15,16};

printf("%d ,%d ,%d\n",sizeof(arr),sizeof(\*arr),sizeof(arr[0]));

}

30) #include<stdio.h>

main()

{

double a[2][3];

printf("%d ",sizeof(a));

printf("%d ",sizeof(a[1]));

printf("%d \n",sizeof(a[1][1]));

}

31) main()

{

int val = 5;

printf(“%d\n”,++val++);

int \*p=&val;

printf(“%d\n”,++\*p++);

}

32) #include<stdio.h>

main()

{

char \*ptr="Hello World";

\*ptr='h';

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

}

33) #include<stdio.h>

main()

{

char \*str="This";

char \*ptr="Program";

str=ptr;

printf("%s %s\n",str,ptr);

}

34) Difference between const char \*p, char const \*p, char \*const p.

a) there is no difference at all

b) first two same, they declare a pointer to a const character the third one declares a

const pointer to a (variable) character

c) the first one declares a pointer to a const character,the last two are same they declare

a const pointer to a(variable) character

d) none of these.

35) #include<stdio.h>

int square (volatile int \*ptr)

{  
 return \*ptr \* \*ptr;

}

void f(char \*k)

{

k++;

k[2]='m';

}

main()

{

char s[]="hello";

f(s);

printf("%c\n",\*(s+2));

}

36) #include<stdio.h>

main()

{

char s[]="man";

int i;

for(i=0;s[i];i++)

printf("\n%c %c %c %c",s[i],\*(s+i),\*(i+s),i[s]);

printf("\n");

}

37) What is array of pointer & pointer to an array and what is the difference.

38) #include<stdio.h>

main()

{

int i,a[]={2,4,6,8,10};

change(a,5);

for(i=0;i<=4;i++)

printf("%d ",a[i]);

}

void change(int \*b,int n)

{

int i;

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

\*(b+1)=\*(b+i)+5;

}

39) #include<stdio.h>

void t1(char \*q)

{

if(\*q!='r')

{

putchar(\*q);

//t1(q++);//segmentation fault

t1(++q);

}

}

main()

{

char \*p;

p="abcder";

t1(p);

}

40) #include<stdio.h>

void f1(int \*,int);

void f2(int \*,int);

void (\*p[2])(int \*,int);

main()

{

int a;

int b;

p[0]=f1;

p[1]=f2;

a=3;

b=5;

p[0](&a,b);

printf("%d\t %d\t",a,b);

p[1](&a,b);

printf("%d\t %d\t",a,b);

}

void f1(int \*p,int q)

{

int temp;

temp=\*p;

\*p=q;

q=temp;

}

void f2(int \*p,int q)

{

int temp;

temp=\*p;

\*p=q;

q=temp;

}

41) #include<stdio.h>

main()

{

int arr[2][2][2]={10,2,3,4,5,6,7,8};

int \*p,\*q;

p=&arr[1][1][1];

q=(int\*)arr;

printf("%d,%d\n",\*p,\*q);

}

42) test is run from command line as ./test Friday Tuesday Sunday

#include<stdio.h>

main(int argc,char \*argv[])

{

printf("%c\n",\*\*++argv);

}

a) t

b) F

c) test

d) Friday

43) #include<stdio.h>

main(int argc,char \*\*argv)

{

if(argc==1)

printf("error\n");

printf("%c ",\*(argv[1]+1));

printf("%c ",(\*(argv+1))[2]);

printf("%c \n",argv[1][2]); }

44) What does the following mean

int (\*ptr)[10];

a) invalid declaration

b) ptr is a pointer to an array of 10 integers

c) ptr is an array of 10 pointers to integers

d) none

45) #include<stdio.h>

main()

{

int a[][3]={1,2,3,4,5,6};

int (\*ptr)[3]=a;

printf("%d %d ",(\*ptr)[1],(\*ptr)[2]);

++ptr;

printf("%d %d \n",(\*ptr)[1],(\*ptr)[2]);

}

46) #include<stdio.h>

main()

{

char \*ptr;

ptr=(char \*)strtok("jan:feb:mar",":");

printf("%s\n",ptr);

do

{

ptr=strtok('\0',":");

if(ptr)

printf("%5s\n",ptr);

}while(ptr);

}

47) #include<stdio.h>

int fun\_def(int a,int b)

{

return a%b;

}

main()

{

int a,b;

int (\*fn\_ptr)(int,int);

a=4;

b=3;

fn\_ptr=fun\_def;

printf("the output is :%d\n",fn\_ptr(a,b));

}

48) #include<stdio.h>

void foo(int b[][3])

{

++b;

b[1][1]=9;

}

main()

{

int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};

foo(a);

printf("%d\n",a[2][1]);

}

49) Which is correct declaration when f is function returning pointer to an array of pointer

to function returning character?

a) char (\*(\*f())[])();

b) char \*(\*f())[];

c) char (\*f()[]());

d) char [\*(\*f())];

50) what would be the output of following program assuming that the array begin at

location 1002 ?

#include<stdio.h>

main()

{

int a[2][3][4]= {

{ 1,2,3,4,

5,6,7,8,

9,1,1,2

},

{

2,4,1,7,

6,7,8,9,

0,0,0,0

}

};

printf("%u %u %u %d\n",a,\*a,\*\*a,\*\*\*a);

}

a) 1002 1002 1002 1002

b) 1002 1002 1002 1

c) 1002 1002 0 1

d) none

51) Memory of 20 bytes is allocated to a string declared as char \*s.what is the value of strlen(s) after copying a string

"Entrance" into that

a) 20

b) 8

c) 9

d) 21

52) #include<stdio.h>

void foo(int a,int b,...)

{

int j;

int \*ptr=&b;

j=0;

while(j<a)

{

printf("%d ",\*ptr);

++j;

++ptr;

}

}

main()

{

foo(1,5);

foo(2,5,6);

}

53) #include<stdio.h>

void f(char \*);

main()

{

f("123");

}

void f(char a[])

{

if(a[1]=='\0')

return;

f(a+1);

f(a+1);

printf("%c ",a[1]);

printf("\n");

}

54) #include<stdio.h>

static unsigned char h[5]={1,2,3,4,5};

main()

{

struct ad{

unsigned short a;

unsigned short b;

};

struct ad \*it;

it=(struct ad \*)h;

printf("%d ",it->a);

printf("%d \n",it->b);

}

55) #include<stdio.h>

int main()

{

static int var=7;

int data;

if(--var)

{

data = main()+var;

printf("%d ",data);

}

}

56) #include<stdio.h>

int foo(int x,int n)

{

int val;

val=1;

if(n>0)

{

if(n%2==1)

val=val\*x;

val=val\*foo(x\*x,n/2);

}

}

main()

{

int r;

r=foo(2,4);

printf("%d\n",r);

}

57) What is memory leak? What functions you call to check memory availability?

How can you avoid memory leaks?

58) if p1 address is 1002 then what will be the addresses for p2 and p3 pointers.

#include<stdio.h>

#include<stdlib.h>

main()

{

int \*p1,\*p2,\*p3;

p1=malloc(10);

printf("p1----> %u\n",(unsigned int)p1);

free(p1);

p2=malloc(5);

printf("p2----> %u\n",(unsigned int)p2);

free(p2);

p3=malloc(5);

printf("p3----> %u\n",(unsigned int)p3);

free(p3);

}

59) #include<stdio.h>

#include<string.h>

void allocate(char \*s)

{

s=malloc(10);

strcpy(s,"hello");

}

60) #include<stdio.h>

#include<malloc.h>

main()

{

int \*p = (int \*)malloc(10);

printf(“p = %u\n”,(unsigned int)p);

int \*q = (int \*)malloc(10);

printf(“q = %u\n”,(unsigned int)q);

free(p);

int \*r = (int \*)malloc(10);

printf(“r = %u\n”,(unsigned int)r);

}

61)What is void pointer?Why can't we perform arithemetic on a void \* pointer?

62) Does an array always get converted to a pointer? What is the difference between arr and &arr? How does one declare a pointer to an entire array?

63) main()

{

printf("%d %d\n",sizeof("string"),strlen("string"));

}

64) #include<stdio.h>

main()

{

int x=20;

{

int x=40;

printf("%d ",x);

}

printf("%d \n",x);

}

65) #include<stdio.h>

void func()

{

int x=0;

static int y=0;

x++;

y++;

printf("%d------%d\n",x,y);

}

66) main()

{

func();

func();

}

67) #include<stdio.h>

char i;

void try1()

{

static char \*ptr="abcde";

i=\*ptr;

printf("%c ",i);

try2(++ptr);

}

void try2(char \*t)

{

static char \*pt;

pt=t+strlen(t)-1;

if(i!=\*pt--)

if(t!=pt)

try1();

}

main()

{

try1();

}

68) what do you mean following code.

static int func(const char c)

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

a) return value of func is a static integer

b) scope of func is limited to the file

c) static cannot be used in the above context

d) none of these

69) which of the following is correct?

a) BSS contains initialized data

b) BSS contains automatic variables

c) BSS contains uninitialized data

d) none of these

70) #include<stdio.h>

main()

{

register int i=10;

int \*ptr;

ptr=&i;

printf("%d\n",\*ptr);

}

71) main()

{

static int i=5;

if(--i)

main();

printf("%d ",i);

}

72) struct employ1

{

char g1;

char g2;

short det;

int roll;

}emp1;

struct employ2

{

char g1;

int roll;

char g2;

short det;

}emp2;

main()

{

printf("%d %d\n",sizeof(emp1),sizeof(emp2));

}

73) #include<stdio.h>

main()

{

struct

{

char a[11];

int i;

}st={"done",10};

printf("%s\n",(&st)->a);

printf("%d\n",(&st)->i);

}

74) #include<stdio.h>

#include<string.h>

struct st

{

int a;

char b;

void \*c;

};

main()

{

struct st \*ptr=NULL;

printf("a:%d\tb:%d\tc:%d\t\*ptr:%d\n",sizeof(ptr->a),sizeof(ptr->b),

sizeof(ptr->c),sizeof(\*ptr));

}

75) How would you find the sizeof structure with out using sizeof operator?

76) #include<stdio.h>

#include<stdlib.h>

struct foo

{

void (\*func)(void);

};

void boo()

{

printf("hello world\n");

}

main()

{

struct foo \*myfoo;

myfoo->func=boo;

myfoo->func();

}

77) what is the size of struct a?

struct a

{

char category :2;

char scheme :4;

};

printf("size= %d\n",sizeof(struct a));

a) 2

b) 1

c) 9

d) none of these

78) #include<stdio.h>

struct st{

int a;

int b;

};

void foo(struct st \*p)

{

char \*pt;

p->a=768;

p->b=128;

pt=(char\*)p;

printf("%d\n",\*++pt);

}

main()

{

struct st ab={128,768};

struct st \*pq=&ab;

foo(pq);

}

79) #include<stdio.h>

main()

{

union

{

struct

{

char c[2];

char ch[2];

}s;

struct

{

int i;

int j;

}st;

}u={12,1,15,1};

printf("%d %d\n",u.st.i,u.st.j);

}

80) #include<stdio.h>

main()

{

struct node

{

int a;

int b;

int c;

};

struct node s={3,5,6};

struct node \*ptr=&s;

printf("%d\n",\*(int \*)ptr);

printf("%d\n",++\*(int \*)ptr);

}

81) #include<stdio.h>

main()

{

unsigned char c;

typedef struct name

{

long a;

int b;

long c;

}r;

r re={3,4,5};

r \*na=&re;

printf("%d",\*(int\*) ( (char\*)na + (unsigned int) & (((struct name \*)0)->b)) );

}

82) union u

{

int i;

char ch;

};

main()

{

union u u1;

u1.i=258;

u1.ch='a';

printf("%d\n",u1.i);

}

83) What are bitfields in structures? What is union? Where does one use unions? What are the

limitaions of unions?

84) Write an example program to show usage of enum's?

85) enum day={jan=1,feb=4,april,may};

what is value of may?

a) 4

b) 5

c) 6

d) 10

86) #include<stdio.h>

#include<string.h>

enum a{NEGATIVE=-1,ZERO,POSITIVE};

void fun(int val)

{

if(val<=NEGATIVE)

printf("negative\n");

else if(val>=POSITIVE)

printf("positive\n");

else

printf("its zero\n");

}

main()

{

fun(-3);

fun(2);

return 0;

}

87) What is macro? Advantages?

88) #include<stdio.h>

int j=0;

main()

{

int i;

printf("i=%d j=%d\n",i,j);

#if((i==0)&&(j==0))

printf("true");

#else

printf("false");

#endif

}

89) #define SWAP(a,b,c) (c t;t=a,a=b,b=t;)

main()

{

int x=10,b=20;

SWAP(x,y,int);

printf("%d %d\n",x,y);

}

a) 10 20

b) 20 10

c) compile time error

d) none

90) #define str(x) #x

#define Xstr(x) str(x)

#define oper multiply

main()

{

char \*opername=Xstr(oper);

printf("%s\n",opername);

}

a) multiply

b) compile time error

c) segmentation fault

d) none

91) #include<stdio.h>

#define ABC() 20

#define XYZ 10

#define NUM ABC()-XYZ

main()

{

int a=10,c;

c=a\*NUM;

printf("%d\n",c);

}

------------------------------------------------------ END ----------------------------------------------------------

Dear Students, if you find any mistakes, please inform to me.

A.Tandava Ramakrishna

Email: ramakrishna@vectorindia.org