Sheet-8 Solution (pointer)

1. main()

{

char \*p1,p2;

printf("%d%d",sizeof(p1),sizeof(p2));

}

**Ans: 4 1 (4 is compiler dependant)**

2. main()

{

printf("%d%d%d",sizeof(char\*),sizeof(int\*),sizeof(float\*));

}

**Ans: 4 4 4**

3. What will be the value of \*ptr1?

#include <stdio.h>

int main()

{

int a = 1, b = 2, c = 3;

int \*ptr1 = &a, \*ptr2 = &b, \*ptr3 = &c;

int \*\*sptr = &ptr1; //-Ref

\*sptr = ptr2;

printf("%d",\*ptr1);

}

**Ans :ptr1 points to b(2)**

4. main()

{

int a=10;

int \*ptr=&a;

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

}

**Ans:1 or garbage 10**

5. main()

{

int a=10;

int \*ptr=&a;

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

}

**Ans: 11 11**

6. main()

{

int a=10;

const int \*ptr=&a;

\*ptr=50;

printf("The changed value of pointed object is%d",\*ptr);

}

**Ans: error: assignment of read-only location ‘\*ptr’**

7. main()

{

int a=10,b=20;

int \*const ptr=&a;

\*ptr=20;

printf("The changed value of pointed object is%d",\*ptr);

ptr=&b;

\*ptr=10;

printf("The changed value of pointed object is%d",\*ptr);

}

**Ans: error: assignment of read-only variable‘ptr’**

8. main()

{

int a=10,b=20;

const int \*const ptr=&a;

\*ptr=20;

printf("The changed value of pointed object is%d",\*ptr);

ptr=&b;

\*ptr=10;

printf("The changed value of pointed object is%d",\*ptr);

}

**Ans:error: assignment of read-only location ‘\*ptr’**

**\*ptr=20;**

**error: assignment of read-only variable ‘ptr’**

**ptr=&b;**

**error: assignment of read-only location ‘\*ptr’**

**\*ptr=10;**

9. main()

{

int \*ptr=10;

printf("The value of pointer is%p",ptr);

}

**Ans: The value of pointer is0xa**

10. main()

{

int \*ptr=0;

printf("The value of pointer is%p",ptr);

}

**Ans: The value of pointer is(nil) or 0**

11 main()

{

int \*ptr1=0;

int \*ptr2=NULL;

if(ptr1==ptr2)

printf("ptr1 becomes a NULL pointer");

else

printf("ptr1 does not become a NULL pointer");

}

**Ans: ptr1 becomes a NULL pointer**

12 What will be the output of the following C code?

#include <stdio.h>

int x = 0;

void main()

{

int \*const ptr = &x;

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

ptr++;

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

}

**Ans:Error**

13. int main(void)  
{  
int target, source;  
int \*m;  
source = 10;  
m = &source;  
target = \*m;  
printf("Value is= %d", target);  
return 0;  
}  
**Ans:- Value is=10**14. main( )  
{  
int i = 3, \*x ;  
float j = 1.5, \*y ;  
char k = 'c', \*z ;  
printf ( "\nValue of i = %d", i ) ;  
printf ( "\nValue of j = %f", j ) ;  
printf ( "\nValue of k = %c", k ) ;  
x = &i ;// address of i=2000  
y = &j ; // address of j=3000  
z = &k ; // address of k=4000  
printf ( "\nOriginal address in x = %u", x ) ;  
printf ( "\nOriginal address in y = %u", y ) ;  
printf ( "\nOriginal address in z = %u", z ) ;  
x++ ;  
y++ ;  
z++ ;  
printf ( "\nNew address in x = %u", x ) ;  
printf ( "\nNew address in y = %u", y ) ;  
printf ( "\nNew address in z = %u", z ) ;  
}  
  
**Ans:-**

**Value of i = 3**

**Value of j = 1.500000**

**Value of k = c**

**Original address in x = 2000**

**Original address in y = 3000**

**Original address in z = 4000**

**New address in x = 2004**

**New address in y = 3004**

**New address in z = 4001**

1. What is the output of this C code?

int main()

{

char \*p = NULL;

char \*q = 0;

if (p)

printf(" p ");

else

printf("nullp");

if (q)

printf("q\n");

else

printf(" nullq\n");

}

**Ans: nullpnullq**

1. What will be the output of the program ?

int main()

{

int i=3, \*j, k;

j = &i;

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

return 0;

}

**Ans :30**

1. What will be the output of the program ?

int main()

{

int x=30, \*y, \*z;

y=&x; /\* Assume address of x is 500 and integer is 4 byte size \*/

z=y;

\*y++=\*z++;

x++;

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

return 0;

}

**ANS: x=31, y=504, z=504**

1. What is the output of this C code?

#include <stdio.h>

int x = 0;

void main()

{

int \*ptr = &x;

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

x++;

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

}

**ANS: Same address**

1. What will be the output of the program If the integer is 4bytes long?

int main()

{

int \*\*\*r, \*\*q, \*p, i=8;

p = &i;

q = &p;

r = &q;

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

return 0;

}

**ANS: 8, 8, 8**

1. What will be the output of the program If the integer is 4bytes long?

int main()

{

int \*\*\*r, \*\*q, \*p, i=8; //adresss of i is 4000, q is 4004, r is 4016

p = &i;

q = &p;

r = &q;

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

return 0;

}

**ANS: 4000, 4000, 4000**

1. What is the output of this C code?

#include <stdio.h>

void main()

{

int x = 0;

int \*ptr = &5;

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

}

**ANS: Compile time error**

1. What is the output of this C code?

#include <stdio.h>

void main()

{

int x = 0;

int \*ptr = &x;

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

}

**ANS:0**

1. What will be the output of the program ?

int main()

{

char \*str;

str = "%d\n";

str++;

str++;

printf(str-2, 300);

return 0;

}

**ANS:300**

1. What will be the output of the program ?

int main()

{

char \*p;

p="hello";

printf("%s\n", \*&\*&p);

return 0;

}

**Ans:- hello**

25. void main()

{

int a=5,\*ptr;

ptr=&a;

printf(“input no”);

scanf("%d",ptr); //enter 10

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

}**Output:- 10, 10**

26. main()

{

int \*ptr;

printf("input no");

scanf("%d",ptr); // enter 10

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

}

**output: -input no10**

**Segmentation fault (core dumped)**

27. void fun(int y)  
{  
    y = 30;  
}  
 int main()  
{  
  int y = 20;  
  fun(y);  
  printf("%d", y);  
  return 0;  
}

**Ans:- 20**28 void f(int \*p, int \*q)  
{  
  p = q;  
  \*p = 2;  
}  
int i = 0, j = 1;  
int main()  
{  
  f(&i, &j);  
  printf("%d %d \n", i, j);  
  return 0;  
}  
**Ans:- 0 2**29# include <stdio.h>  
void fun(int \*ptr)  
{  
    \*ptr = 30;  
}  
int main()  
{  
  int y = 20;  
  fun(&y);  
  printf("%d", y);  
  return 0;  
}  
**Ans:- 30**30.int main()

{

int \*ptr;

int x;

ptr = &x;

\*ptr = 0;

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

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

\*ptr += 5;

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

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

(\*ptr)++;

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

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

return 0;

}

**Ans:- x = 0**

**\*ptr = 0;**

**x = 5**

**\*ptr = 5**

**x = 6**

**\*ptr = 6**

31.int main()  
{  
   int a;  
   char \*x;  
   x = (char \*) &a;  
   a = 512;  
   x[0] = 1;  
   x[1] = 2;  
   printf("%d\n",a);    
   return 0;  
}  
  
**Ans:- 513**  
32. int main()  
{  
 char \*ptr;

ptr = "GeeksQuiz";  
 printf("%cn", \*&\*&\*ptr);  
 return 0;  
}

**Ans:- G**

33.#include <stdio.h>  
/\* function declaration \*/  
void swap(int x, int y);  
int main ()  
{  
/\* local variable definition \*/  
int a = 100;  
int b = 200;  
printf("Before swap, value of a : %d\n", a );  
printf("Before swap, value of b : %d\n", b );  
/\* calling a function to swap the values \*/  
swap(a, b);  
printf("After swap, value of a : %d\n", a );  
printf("After swap, value of b : %d\n", b );  
return 0;  
}

void swap(int a,int b)

{

int temp;

temp=a;

a=b;

b=temp;

}

**Ans:- Before swap, value of a : 100**

**Before swap, value of b : 200**

**After swap, value of a : 100**

**After swap, value of b : 200**

34#include <stdio.h>  
/\* function declaration \*/  
void swap(int \*x, int \*y);  
int main ()  
{  
/\* local variable definition \*/  
int a = 100;  
int b = 200;  
printf("Before swap, value of a : %d\n", a );  
printf("Before swap, value of b : %d\n", b );  
/\* calling a function to swap the values.  
\* &a indicates pointer to a ie. address of variable a and  
\* &b indicates pointer to b ie. address of variable b.  
\*/  
swap(&a, &b);  
printf("After swap, value of a : %d\n", a );  
printf("After swap, value of b : %d\n", b );  
return 0;  
}  
void swap(int \*a,int \*b)

{

int temp;

temp=\*a;

\*a=\*b;

\*b=temp;

}

**Ans:- Before swap, value of a : 100**

**Before swap, value of b : 200**

**After swap, value of a : 200**

**After swap, value of b : 100**

35. int f(int x, int \*py, int \*\*ppz)  
{  
  int y, z;  
  \*\*ppz += 1;  
   z  = \*\*ppz;  
  \*py += 2;  
   y = \*py;  
   x += 3;  
   return x + y + z;  
}     
void main()  
{  
   int c, \*b, \*\*a;  
   c = 4;  
   b = &c;  
   a = &b;  
   printf("%d ", f(c, b, a));  
   return 0;  
}  
**Ans:- 19**

36. #include<stdio.h>  
int main()  
{  
    int a = 12;  
    void \*ptr = (int \*)&a;  
    printf("%d", \*ptr);  
    return 0;  
}  
**Ans:- error**

main.c: In function ‘main’:

main.c:6:18: warning: dereferencing ‘void \*’ pointer

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

^~~~

main.c:6:5: error: invalid use of void expression

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

^~~~~~

37#include<stdio.h>

int main()

{

int a = 12;

void \*ptr = (int \*)&a;

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

return 0;

}

**Ans:- Address**

38#include<stdio.h>

int main()

{

int a = 12;

void \*ptr = (int \*)&a;

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

return 0;

}

**Ans:- 12**  
  
39. void mystery(int \*ptra, int \*ptrb)  
{  
   int \*temp;  
   temp = ptrb;  
   ptrb = ptra;  
   ptra = temp;  
}  
int main()  
{  
    int a=2016, b=0, c=4, d=42;  
    mystery(&a, &b);  
    if (a < c)  
       mystery(&c, &a);  
    mystery(&a, &d);  
    printf("%d\n", a);  
}  
**Ans:- 2016**

40. void f(int\* p, int m)  
{  
    m = m + 5;  
    \*p = \*p + m;  
    return;  
}  
void main()  
{  
    int i=5, j=10;  
    f(&i, j);  
    printf("%d", i+j);  
}

**Ans:- 30**  
41. int main()  
{  
   int a = 300;      
   char \*b = (char \*)&a;  
   \*++b = 2;  
   printf("%d ",a);  
   return 0;  
}

**Ans:- 556**