1. Which header file should be included to use functions like `malloc()` and `calloc()`?

   A. memory.h

   B. stdlib.h

   C. string.h

   D. dos.h

   Answer: Option **B**
   Explanation:
   No answer description available for this question. Let us discuss.
   View Answer Discuss in Forum Workspace Report

---

2. What function should be used to free the memory allocated by `calloc()` ?

   A. dealloc();

   B. malloc(variable_name, 0)

   C. free();

   D. memalloc(variable_name, 0)

   Answer: Option **C**
   Explanation:
   No answer description available for this question. Let us discuss.
   View Answer Discuss in Forum Workspace Report

---

3. How will you free the memory allocated by the following program?

```c
#include<stdio.h>
#include<stdlib.h>
#define MAXROW 3
#define MAXCOL 4

int main()
{
    int **p, i, j;
    p = (int **) malloc(MAXROW * sizeof(int*));
    return 0;
}
```

   A. memfree(int p);

   B. dealloc(p);

   C. malloc(p, 0);

   D. free(p);

   Answer: Option **D**
   Explanation:
   No answer description available for this question. Let us discuss.

View Answer Discuss in Forum Workspace Report

4. Specify the 2 library functions to dynamically allocate memory?

   A. `malloc()` and `memalloc()`

   B. `alloc()` and `memalloc()`

   C. `malloc()` and `calloc()`

   D. `memalloc()` and `faralloc()`

   Answer: Option **C**

1. What will be the output of the program?

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int *p;
    p = (int *)malloc(20); /* Assume p has address of 1314 */
    free(p);
    printf("%u", p);
    return 0;
}
```

   A. 1314

   B. Garbage value

   C. 1316

   D. Random address

   Answer: Option **A**
   Explanation:
   No answer description available for this question. Let us discuss.
   View Answer Discuss in Forum Workspace Report

2. What will be the output of the program (16-bit platform)?

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int *p;
    p = (int *)malloc(20);
    printf("%d\n", sizeof(p));
    free(p);
    return 0;
}
```

A. 4

B. 2

C. 8

D. Garbage value

Answer: Option **B**
Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

3. What will be the output of the program?

```
#include<stdio.h>
#include<string.h>

int main()
{
    char *s;
    char *fun();
    s = fun();
    printf("%s\n", s);
    return 0;
}
char *fun()
{
    char buffer[30];
    strcpy(buffer, "RAM");
    return (buffer);
}
```

A. 0xffff

B. Garbage value

C. 0xffee

D. Error

Answer: Option **B**
Explanation:
The output is unpredictable since `buffer` is an auto array and will die when the control go back to `main`. Thus `s` will be pointing to an array , which not exists.
View Answer Discuss in Forum Workspace Report

4. What will be the output of the program?

```
#include<stdio.h>
#include<stdlib.h>
```

```
int main()
{
    union test
    {
        int i;
        float f;
        char c;
    };
    union test *t;
    t = (union test *)malloc(sizeof(union test));
    t->f = 10.10f;
    printf("%f", t->f);
    return 0;
}
```

A. 10

B. Garbage value

C. 10.100000

D. Error

Answer: Option **C**
Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

5. Assume integer is 2 bytes wide. How many bytes will be allocated for the following code?

```
#include<stdio.h>
#include<stdlib.h>
#define MAXROW 3
#define MAXCOL 4

int main()
{
    int (*p)[MAXCOL];
    p = (int (*) [MAXCOL])malloc(MAXROW *sizeof(*p));
    return 0;
}
```

A. 56 bytes

B. 128 bytes

C. 24 bytes

D. 12 bytes

Answer: Option **C**

7. How many bytes of memory will the following code reserve?

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int *p;
    p = (int *)malloc(256 * 256);
    if(p == NULL)
        printf("Allocation failed");
    return 0;
}
```

A. 65536

B. Allocation failed

C. Error

D. No output

Answer: Option **B**

1. Point out the error in the following program.

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int *a[3];
    a = (int*) malloc(sizeof(int)*3);
    free(a);
    return 0;
}
```

A. Error: unable to allocate memory

B. Error: We cannot store address of allocated memory in `a`

C. Error: unable to free memory

D. No error

Answer: Option **B**
Explanation:
We should store the address in `a[i]`
View Answer Discuss in Forum Workspace Report

---

2. Point out the error in the following program.

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    char *ptr;
    *ptr = (char)malloc(30);
    strcpy(ptr, "RAM");
    printf("%s", ptr);
    free(ptr);
    return 0;
}
```

A. Error: in `strcpy()` statement.

B. Error: in `*ptr = (char)malloc(30);`

C. Error: in `free(ptr);`

D. No error

Answer: Option **B**
Explanation:
Answer: `ptr = (char*)malloc(30);`

1. Point out the correct statement will let you access the elements of the array using 'p' in the following program?

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int i, j;
    int(*p)[3];
    p = (int(*)[3])malloc(3*sizeof(*p));
    return 0;
}
```

A.

```
for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
        printf("%d", p[i+j]);
}
```

B.

```
for(i=0; i<3; i++)
    printf("%d", p[i]);
```

C.

```
for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
        printf("%d", p[i][j]);
}
```

D.
```
for(j=0; j<3; j++)
    printf("%d", p[i][j]);
```

Answer: Option **C**
Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

2. Which of the following statement is correct prototype of the `malloc()` function in c ?

   A. int* malloc(int);

   B. char* malloc(char);

   C. unsigned int* malloc(unsigned int);

   D. void* malloc(size_t);

Answer: Option **D**
Explanation:
No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

3. Point out the correct statement which correctly free the memory pointed to by 's' and 'p' in the following program?

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
    struct ex
    {
        int i;
        float j;
        char *s
    };
    struct ex *p;
    p = (struct ex *)malloc(sizeof(struct ex));
    p->s = (char*)malloc(20);
    return 0;
}
```

   A. free(p); , free(p->s);

   B. free(p->s); , free(p);

   C. free(p->s);

   D. free(p);

Answer: Option **B**
Explanation:

No answer description available for this question. Let us discuss.
View Answer Discuss in Forum Workspace Report

---

4. Point out the correct statement which correctly allocates memory dynamically for 2D array following program?

```c
#include<stdio.h>
#include<stdlib.h>

int main()
{
    int *p, i, j;
    /* Add statement here */
    for(i=0; i<3; i++)
    {
        for(j=0; j<4; j++)
        {
            p[i*4+j] = i;
            printf("%d", p[i*4+j]);
        }
    }
    return 0;
}
```

A. p = (int*) malloc(3, 4);

B. p = (int*) malloc(3*sizeof(int));

C. p = malloc(3*4*sizeof(int));

D. p = (int*) malloc(3*4*sizeof(int));

Answer: Option **D**