1.

```
#include <stdio.h>
#include <string.h>
struct demo
{
	unsigned int _1 : 6;
	signed int _2 : 7;
	unsigned int _3 : 6;
};
int main(void )
{
	typedef struct demo DEMO;

	DEMO _d;

	_d._1 = 88 ;
	_d._2 = 77 ;
	_d._1 = 58 ;

	printf( " %d %d %d \t",_d._1 ,_d._2, _d._3 );

	_d._1 = 92 ;
	_d._2 = 96 ;
	_d._2 = 99 ;

	printf( " %d %d %d \n",_d._1 ,_d._2, _d._3 );

	return 0;
}
```

A. 57 -50 -1 29 -28 -1
B. 58 -50 0 28 -28 0
C. 58 -51 0 28 -29 0
D. 57 -51 -1 28 -27 -1
E. Error

Answer: C

2.

```
#include<stdio.h>
#pragma pack(1)
typedef union result
{
	float per;
	char grade[4];
}RESULT;
typedef struct student
{
	int rollno;
	short int std;
	char name[12];
	char div;
	RESULT res;
}STUDENT;
typedef struct node
{
	struct test *prev;
	STUDENT data;
	struct test *next;
}node_t;

int main(void)
{
	printf("\n size of node=%d", sizeof(node_t));

	return 0;
}
```

what will be the size of struct node? Note: consider 64 bit compiler

A. size of node=44
B. size of node=43
C. size of node=40
D. size of node=39
E. ERROR

Answer: D

3.

```
#include<stdio.h>
#pragma pack(1)

struct time
{
	int ss:9;
	int mm:8;
	int hh:6;
};
int main(void)
{
	struct time t1;
	printf("%d %d", sizeof(t1), sizeof(t1.ss)+ sizeof(t1.mm) );
	return 0;
}
```

A. 3 2
B. Compiler error
C. Runtime error
D. 4 2
E. 4 3

Answer: B

4.

```
#include <stdio.h>
int main(void)
{
	FILE *fp = stdout;
	char str[] = "\?\?!\n?";
	if( (fp=fopen("sunbeam.txt", "w"))!=NULL)
		fprintf(fp,"%s", str);
	else
		fputs(str, fp) ;
	fclose(fp);
	return 0;
}
```

A. Compile Time Error
B. Run Time Error
C. ??!
D. \?\?!\n?
E. ??!
   ?

Answer: E

5.

```
#include <stdio.h>
int main(void)
{
	FILE *fp = stdout;
	char str[] = "\\t SunBeam \\0 \\n \0 Pune";
	if( (fp=fopen("sunbeam.txt", "w"))!=NULL)
		fprintf(fp,"%s", str);
	else
		fprintf(fp,"%s", "unable to write in file");
	fclose(fp);

	return 0;
}
```

A. \t SunBeam \0 \n
B. \\t SunBeam \\0 \\n \0 Pune
C. \t SunBeam \0 \n \0 Pune
D. Error
E. SunBeam \\0
    \0 Pune

Answer: A